

Titre: Étude d'un modèle génératif pour l'analyse en temps réel de trajectoires bidimensionnelles bruitées
Title:

Auteur: Pier-Olivier Genest
Author:

Date: 2008

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Genest, P.-O. (2008). Étude d'un modèle génératif pour l'analyse en temps réel de trajectoires bidimensionnelles bruitées [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/8254/>
Citation:

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/8254/>
PolyPublie URL:

Directeurs de recherche:
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE D'UN MODÈLE GÉNÉRATIF POUR L'ANALYSE EN TEMPS RÉEL
DE TRAJECTOIRES BIDIMENSIONNELLES BRUITÉES

PIER-OLIVIER GENEST
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE, APPRENTISSAGE MACHINE)

MAI 2008



Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-41558-0

Our file Notre référence

ISBN: 978-0-494-41558-0

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ÉTUDE D'UN MODÈLE GÉNÉRATIF POUR L'ANALYSE EN TEMPS RÉEL
DE TRAJECTOIRES BIDIMENSIONNELLES BRUITÉES

présenté par: GENEST Pier-Olivier

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. HURTEAU Richard, Ph.D., président

M. BRAULT Jean-Jules, Ph.D., membre et directeur de recherche

M. GAGNON Michel, Ph.D., membre

*À ma petite famille,
sans laquelle tous ces efforts feraient bien peu de sens.*

REMERCIEMENTS

Je tiens à remercier mon directeur, Jean-Jules Brault, pour sa disponibilité et son support lors des moments difficiles, ses conseils judicieux et pour sa confiance en mes capacités.

Je suis reconnaissant envers le CRSNG pour l'octroi d'une bourse de maîtrise à incidence industrielle qui m'a permis de réaliser ces études.

Merci à mes collègues de travail et plus particulièrement, Emerson Nérat, pour leur compréhension et leurs encouragements.

Un merci tout spécial à ma belle-maman, Michèle Barry, pour le temps investi dans la relecture et la correction des chapitres.

Merci aussi à ma chère sœur, Sophie-Emmanuelle, sans qui ce manuscrit aurait dû être rédigé à la main...

Un gros merci à tout le reste de ma famille et plus particulièrement à mes parents pour leur soutien moral et financier tout au long de cette aventure beaucoup plus longue que prévue.

Je ne saurais comment remercier suffisamment mon épouse Valérie pour son inestimable patience, pour sa compréhension face aux situations plus difficiles dans lesquelles je nous ai embarqués et pour la détermination dont elle a fait preuve depuis toujours. Merci de m'avoir encouragé lorsque je m'apprêtais à flancher, merci de m'avoir botté en dehors du lit, merci de m'avoir aidé à rédiger, merci de m'aimer si fort, merci d'être l'épouse et la mère que tu es.

Finalement, merci à mon petit Arnaud pour tous les sourires que tu m'as faits au moment où j'en avais tant besoin...

RÉSUMÉ

Introduction : Que ce soit par surveillance vidéo, par communication cellulaire, par géolocalisation ou par réseaux de senseurs, plusieurs systèmes sont en mesure de fournir les positions d'agents mobiles en temps réel. **Objectif :** Ce travail consiste à élaborer, dans le paradigme des modèles graphiques probabilistes (PGM) et plus spécifiquement des réseaux bayésiens (BN), un modèle d'activité capable d'analyser en temps réel des trajectoires bidimensionnelles effectuées par un agent mobile autonome.

Méthodologie : Un HHMM, la version hiérarchique d'un modèle markovien caché (HMM), est étudié et testé dans trois contextes d'utilisation afin d'évaluer sa capacité à reconnaître les patrons et les sous-patrons de trajectoires bidimensionnelles de déplacement. Le premier contexte consiste en des séquences aléatoires de primitives gaussiennes. Le second contexte consiste en des séquences aléatoires de primitives de type trajectoire bidimensionnelle. Finalement, des trajectoires bidimensionnelles provenant de réelles compétitions *RoboCup* sont utilisées dans le troisième contexte d'application. Un modèle de segmentation en-ligne basé sur PPCA, la version probabiliste de l'analyse en composantes principales (PCA), est aussi utilisé pour réduire la charge de calcul sur le HHMM. **Résultats :** Les résultats montrent qu'un HHMM entraîné avec l'algorithme de maximisation de l'espérance (EM) de façon non supervisée reconnaît une séquence de primitives comme le fait un HMM avec des scores près de 100% dans le cas idéal gaussien et 75% dans le cas de trajectoires bidimensionnelles. De plus, un HHMM détecte l'existence de sous-patrons. Le coût de calcul élevé limite toutefois l'utilisation de tels PGMs en temps réel surtout lorsque la topologie de la hiérarchie se complexifie. L'ajout d'un modèle PPCA de segmentation en-ligne permet alors de réduire le coût moyen de calcul du HHMM d'un facteur 10^2 tout en maintenant des scores du même ordre.

Discussion/Conclusion : Le modèle PPCA de segmentation en-ligne

s'avère un compromis rendement vs coût de calcul optimal. En effet, le modèle atteint un rendement 2 à 4 fois supérieur au modèle PCA de base pour le double du coût de calcul. Le HHMM étant d'une complexité supérieure au HMM, la malédiction de la dimensionnalité rend l'apprentissage difficile. Les résultats montrent toutefois qu'une régularisation convenable rend le modèle accessible et approprié pour la reconnaissance de trajectoires bidimensionnelles. Le modèle d'activité obtenu semble être une base intéressante pour d'éventuels développements sur les stratégies de planification, la prédiction de trajectoires et la détection d'anomalies.

ABSTRACT

Introduction: Video surveillance, cellular telecommunication, global positioning systems and wireless sensor networks are just a few examples among many others of monitoring systems providing real time location data.

Objective: The purpose of this work is to develop an activity model in the paradigm of probabilistic graphical models (PGM) and Bayesian networks (BN) to analyze in real time two-dimensional trajectories observed from the displacement of autonomous mobile agents.

Method: An HHMM is the hierarchical version of the hidden Markov model (HMM) and is studied and tested here within three contexts of application in order to determine its ability to recognize patterns and sub-patterns of displacement from two-dimensional trajectories. The first context of application resumes to random sequences of Gaussian primitives. The second context of application resumes to random sequences of primitives created from two-dimensional trajectory samples. Finally, two-dimensional trajectories of displacement from real RoboCup competitions are used in the third context of application. An online segmentation model based on PPCA, the probabilistic version of principal component analysis (PCA), is then used to reduce the computation cost involving the HHMM. **Results:** It is shown that when trained in an unsupervised fashion with the expectation maximization (EM) algorithm, the HHMM as well as the HMM lead to a recognition score of almost 100% in the idealistic case of Gaussian primitives and to a score of 75% in the case of two-dimensional trajectories. The HHMM also has the ability to recognize sub-patterns. However such PGMs might not be tractable in real time because of the high computation cost introduced by potentially complex hierarchical topologies. The computation cost of the HHMM can be reduced by a factor of 10^2 with the addition of an online PPCA segmentation model, keeping recognition scores approximately the same.

Discussion/Conclusion: The online PPCA segmentation model stands as

the best trade-off between segmentation efficiency and computation cost. Indeed, the online PPCA segmentation model is 2 to 4 times more efficient than the basic PCA model for the double of the cost. Since the HHMM has a complexity level greater than the HMM the curse of the dimensionality leads to overfitting problems. Results show however that the model is relevant and appropriate for two-dimensional trajectory recognition if a proper regularization is performed. Finally, the achievements on the activity model presented in this work might provide an interesting framework for developments regarding plan recognition, trajectory prediction and anomaly detection.

TABLE DES MATIÈRES

REMERCIEMENTS	V
RÉSUMÉ	VI
ABSTRACT	VIII
LISTE DES TABLEAUX	XIV
LISTE DES FIGURES	XVI
LISTE DES ANNEXES	XIX
LISTE DES SIGLES ET ABRÉVIATIONS	XX
LISTE DES SYMBOLES	XXIII
AVANT-PROPOS	XXV
CHAPITRE 1 : « LES RÈGLES DU JEU »	1
1.1 Mise en situation	1
1.2 Où se situe l'intérêt ?	2
1.3 Pourquoi limiter l'analyse aux trajectoires bidimensionnelles ?	3
1.4 Qu'est-ce que le modèle d'activité?	4
1.5 Quelques considérations	8
1.5.1 Les interactions avec les autres agents	8
1.5.2 L'environnement	9
1.5.3 Les buts	10
1.5.4 Les périodes récurrentes	11
1.5.5 Les anomalies	11
1.5.6 La résolution	12
1.5.7 Le traitement en temps réel	14
1.5.8 Le bruit	14
1.5.9 Les considérations retenues	15
1.6 Pourquoi les modèles graphiques probabilistes ?	15
1.7 La problématique et les objectifs	19
1.8 Une approche non supervisée	20
1.9 Pourquoi RoboCup ?	21

1.10 La plateforme de travail	22
CHAPITRE 2 : « L'AVANT-MATCH »	24
2.1 Point de vue de la planification	25
2.1.1 Les interactions multi-agent	27
2.2 Point de vue des modèles comportementaux et d'activité	28
2.2.1 Le cas multi-agent	30
2.2.2 La reconnaissance visuelle	30
2.2.3 La navigation et le mappage de l'environnement	31
2.3 Dans le cadre des compétitions de soccer multi-agent	31
2.3.1 Le paradigme RoboCup	32
2.3.2 L'analyse de trajectoires	34
CHAPITRE 3 : « LA FORMATION DE L'ÉQUIPE »	35
3.1 La première approche considérée	35
3.2 L'alternative gagnante	38
3.3 Les distinctions du modèle envisagé	41
CHAPITRE 4 : « L'ÉCHAUFFEMENT »	45
4.1 Modèle d'activité multi-résolution	45
4.1.1 Synchronisme des sources de données	47
4.1.2 Format des données	47
4.2 L'espace des primitives	48
4.2.1 Le prétraitement : augmenter la dimensionnalité	48
4.3 Le modèle graphique probabiliste (PGM)	51
4.3.1 Le modèle markovien caché (HMM)	51
4.3.2 Le modèle markovien caché hiérarchique (HHMM)	55
4.4 Le modèle de segmentation en-ligne	60
4.4.1 Technique de segmentation PCA (critère dimensionnalité)	61
4.4.2 Technique de segmentation PCA (critère base vectorielle)	63
4.4.3 Technique de segmentation PPCA	64
CHAPITRE 5 : « LE PLAN DE MATCH »	67
5.1 Les trois contextes d'application	67
5.1.1 Séquences (abstraites) de primitives gaussiennes	67

5.1.2	Séquences (aléatoires) de primitives 2D	69
5.1.3	Trajectoires réelles (RoboCup)	71
5.2	Méthodologie de test : PGM seul	73
5.2.1	Séquences (abstraites) de primitives gaussiennes	74
5.2.2	Séquences (aléatoires) de primitives 2D	75
5.3	Méthodologie de test : PGM avec modèle de segmentation en-ligne	75
5.3.1	Séquences (abstraites) de primitives gaussiennes	76
5.3.2	Séquences (aléatoires) de primitives 2D	78
5.3.3	Trajectoires réelles (RoboCup)	79
5.3.4	Coût de calcul temps réel	80
CHAPITRE 6 : « LA 1^{ÈRE} DEMIE »		82
6.1	Validation du PGM	82
6.1.1	Séquences (abstraites) de primitives gaussiennes	83
6.1.2	Séquences (aléatoires) de primitives 2D	85
CHAPITRE 7 : « LA 2^{ÈME} DEMIE »		90
7.1	Validation du modèle de segmentation en-ligne	90
7.1.1	Séquences (abstraites) de primitives gaussiennes	91
7.1.2	Séquences (aléatoires) de primitives 2D	96
7.1.3	Trajectoires réelles (RoboCup)	99
7.1.4	Coût de calcul temps réel	103
7.2	Validation du PGM avec modèle de segmentation en-ligne	104
7.2.1	Séquences (abstraites) de primitives gaussiennes	104
7.2.2	Séquences (aléatoires) de primitives 2D	107
7.2.3	Trajectoires réelles (RoboCup)	110
7.2.4	Coût de calcul temps réel	113
CHAPITRE 8 : « LES ECHOS DE VESTIAIRE »		116
8.1	Les principales contributions	116
8.2	Le rendement et les limites du PGM	116
8.2.1	Le nombre de paramètres	117
8.2.2	Les primitives gaussiennes	118
8.2.3	Anomalies et traitement en lots	118
8.2.4	Description non supervisée des activités	119

8.2.5	Traitement en temps réel	120
8.3	Le rendement et les limites du modèle de segmentation en-ligne	120
8.3.1	Robustesse face à la dimensionnalité	121
8.3.2	Limitation quant à la résolution	122
8.4	Avenues potentielles pour le modèle d'activité	123
	CONCLUSION	125
	RÉFÉRENCES	127
	ANNEXE 1	132
	Concepts de base sur les réseaux bayésiens	132
	ANNEXE 2	135
	Analyse en composantes principales (PCA)	135
	Technique de segmentation PCA (critère dimensionnalité)	140
	Technique de segmentation PCA (critère base vectorielle)	141
	Technique de segmentation PPCA	142
	Définition des modèles de segmentation en-ligne en terme de leurs hyper-paramètres	144
	ANNEXE 3	146
	Schéma Simulink des opérations de prétraitement sur la dimensionnalité	146
	ANNEXE 4	148
	Validation du PGM seul : résultats complémentaires	148

LISTE DES TABLEAUX

Tableau 4.1 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche continue C1.	50
Tableau 4.2 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche discrète C2.	50
Tableau 4.3 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche logique C3.	51
Tableau 6.1 : Scores des deux PGMs dans un espace de primitives gaussiennes à 3 dimensions.	83
Tableau 6.2 : Scores des deux PGMs dans un espace de primitives gaussiennes à 50 dimensions.	84
Tableau 6.3 : Scores des deux PGMs dans un espace à 52 dimensions (primitives de type trajectoire).	88
Tableau 7.1 : Rendement de divers modèles de segmentation en-ligne et effet du nombre de composantes principales effectives dans un espace à 70 dimensions.	92
Tableau 7.2 : Rendement de divers modèles de segmentation en-ligne dans un espace à 25 dimensions.	93
Tableau 7.3 : Rendement moyen des meilleurs modèles de segmentation.	97
Tableau 7.4 : Statistiques descriptives sur la dynamique des trois trajectoires bidimensionnelles considérées.	100
Tableau 7.5 : Dérivée globale du temps de calcul.	104
Tableau 7.6 : Scores des trois PGMs dans un espace de primitives gaussiennes à 3 dimensions.	105
Tableau 7.7 : Scores des trois PGMs dans un espace de primitives gaussiennes à 50 dimensions.	106

Tableau 7.8 : Scores des trois PGMs dans un espace à 52 dimensions (primitives de type trajectoire).	108
--	-----

Tableau 7.9 : Coûts de calcul moyen et maximal typique de chaque PGM.	115
---	-----

LISTE DES FIGURES

Figure 1.1 : Représentation dans le plan de deux trajectoires bidimensionnelles.	4
Figure 1.2 : Représentations d’une même trajectoire à deux résolutions spatiales distinctes.	12
Figure 1.3 : Représentations d’une même trajectoire à deux résolutions temporelles distinctes.	13
Figure 1.4 : Une partie dans le cadre de RoboCup Simulation League représentée à l’aide du module de relecture.	22
Figure 3.1 : Représentation du BN statique intégrant un modèle d’observation et de « genre prédictif » en un seul BN.	36
Figure 3.2 : BN représentant le modèle d’observation.	37
Figure 3.3 : BN représentant le modèle de « genre prédictif ».	37
Figure 3.4 : Représentation schématique de plusieurs HMMs imbriqués les uns dans les autres afin de former une hiérarchie.	38
Figure 3.5 : Exemple du genre d’analyse qui pourrait être effectuée par un modèle hiérarchique.	39
Figure 3.6 : Influences entre le but, l’action et l’état de l’agent.	41
Figure 3.7 : Représentation schématique d’une hiérarchie de buts/actions/ états.	43
Figure 4.1 : Les trois couches de représentation des diverses résolutions spatiales d’une position dans le plan.	46
Figure 4.2 : L’interface dédiée convertit le format de données des fichiers RoboCup dans le format (t, Id, x, y, c1, c2, ..., ck) requis.	48
Figure 4.3 : Représentation schématique de la tâche de filtrage.	53
Figure 4.4 : Représentation schématique de la tâche de prédiction.	53
Figure 4.5 : Représentation schématique du lissage à retard fixe (a) et du lissage à intervalle fixe (b).	54

Figure 4.6 : Représentation sous forme de treillis de toutes les séquences possibles et de la séquence la plus vraisemblable.	55
Figure 4.7 : Représentation DBN d'un HHMM à 3 niveaux.	57
Figure 4.8 : HHMM à 3 niveaux dépendant uniquement du niveau parent.	57
Figure 4.9 : Diagramme de transition représentant la topologie d'un HHMM à trois niveaux.	59
Figure 5.1 : Quatre séquences d'environ 200 échantillons provenant de primitives gaussiennes tridimensionnelles.	69
Figure 5.2 : Sept trajectoires bidimensionnelles d'environ 200 échantillons provenant de primitives de type trajectoire disimensionnelle.	70
Figure 5.3 : Sept trajectoires bidimensionnelles d'environ 200 échantillons provenant de primitives de type trajectoire disimensionnelle.	71
Figure 5.4 : Trajectoire bidimensionnelle d'un joueur de soccer robotisé durant une partie complète d'une compétition RoboCup.	72
Figure 5.5 : Trajectoire bidimensionnelle d'un gardien de but robotisé durant une partie complète d'une compétition RoboCup.	72
Figure 5.6 : Trajectoire bidimensionnelle du ballon durant une partie complète d'une compétition RoboCup.	73
Figure 6.1 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à trois dimensions.	85
Figure 6.2 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à 50 dimensions.	86
Figure 6.3 : Sur-apprentissage du modèle d'observation d'un HHMM et compensation du premier niveau.	89
Figure 7.1 : Trame des coupures détectées par les modèles de segmentation en ligne pour une séquence de primitives gaussiennes en 70 dimensions.	95
Figure 7.2 : Séquence de type trajectoire bidimensionnelle.	98

- Figure 7.3 : Trame des coupures d'une séquence de type trajectoire bidimensionnelle provenant du patron 5-1-6. 99
- Figure 7.4 : Séquence de six primitives détectées à partir de la trajectoire bidimensionnelle de l'agent mobile joueur. 101
- Figure 7.5 : Évolution de la distance de Mahalanobis moyenne pour les 1500 premiers échantillons de la trajectoire bidimensionnelle des agents mobiles gardien de but (en haut) et joueur (en bas). 102
- Figure 7.6 : Temps de calcul des trois types de modèle de segmentation en-ligne sur une trajectoire bidimensionnelle de 3000 échantillons. 103
- Figure 7.7 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à 50 dimensions. 107
- Figure 7.8 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives à 52 dimensions obtenues à partir de trajectoires 2D générées. 109
- Figure 7.9 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives à 52 dimensions obtenues à partir d'une trajectoire bidimensionnelle observée dans une partie RoboCup. 110
- Figure 7.10 : Deux primitives aisément détectées par le modèle d'activité représentant une pause (une touche) en bordure du terrain. 112
- Figure 7.11 : Temps de calcul de quatre PGMs sur une trajectoire bidimensionnelle de 700 échantillons. 113
- Figure 7.12 : Temps de calcul de quatre PGMs sur une trajectoire bidimensionnelle de 700 échantillons. 114
- Figure 8.1 : Représentation de deux trajectoires issues des patrons 3-1-4 et 5-1-6 avec un agrandissement sur la primitive centrale. 124

LISTE DES ANNEXES

ANNEXE 1	132
ANNEXE 2	135
ANNEXE 3	146
ANNEXE 4	148

LISTE DES SIGLES ET ABRÉVIATIONS

AHMM	Modèle markovien caché abstrait (<i>Abstract hidden markov model</i>)
BN	Réseau bayésien (<i>Bayesian network</i>)
BNT	Librairie Matlab sur les réseaux bayésiens (<i>Bayesian network toolbox</i>)
CPD	Distribution de probabilité conditionnelle (<i>Conditional probability distribution</i>)
CPT	Table de probabilité conditionnelle (<i>Conditional probability table</i>)
DAG	Graphe orienté acyclique (<i>Directed acyclic graphe</i>)
DBN	Réseau bayésien dynamique (<i>Dynamical bayesian network</i>)
EM	Maximisation de l'espérance (<i>Expectation maximization</i>)
FSM	Machine à états finis (<i>Finite state machine</i>)
HHMM	Modèle markovien caché hiérarchique (<i>Hierarchical hidden markov model</i>)
HMM	Modèle markovien caché (<i>Hidden markov model</i>)
HSOM	Réseaux neuronaux auto-organiseurs hiérarchiques (<i>Hierarchical self organizing map</i>)

KFM	Filtre de Kalman (<i>Kalman filter model</i>)
LCSS	Sous-séquence commune la plus longue (<i>Longest common subsequence</i>)
LDS	Système linéaire dynamique (<i>Linear dynamical system</i>)
MDP	Processus markovien de décision (<i>Markov decision process</i>)
MPE	Explication la plus probable (<i>Most probable explanation</i>)
NBC	Classificateur bayésien naïf (<i>Naive Bayesian classifier</i>)
N-gram	Approche grammaticale à N états
PCA	Analyse en composantes principales (<i>Principal component analysis</i>)
PCFG	Grammaires probabilistes non contextuelles (<i>Probabilistic context-free grammars</i>)
PGM	Modèle graphique probabiliste (<i>Probabilistic graphical model</i>)
POMPD	Processus markovien de décision partiellement observable (<i>Partially observable markov decision process</i>)
PPCA	Analyse probabiliste en composantes principales (<i>Probabilistic principal component analysis</i>)
PSDG	Grammaires probabilistes dépendantes à un état (<i>Probabilistic state-dependent grammars</i>)
RFID	Identification par onde radiofréquence

	<i>(Radio frequency identification)</i>
SLAM	Localisation et mappage de l'environnement <i>(Simultaneous localization and mapping)</i>
SLAS	Localisation et mappage sensoriel de l'environnement <i>(Smultaneous localization and surveying)</i>
SSM	Modèle d'espace d'états <i>(State space model)</i>
SVM	Machine à vecteurs de support <i>(Support vectors machine)</i>

LISTE DES SYMBOLES

T, L, M, N	bornes scalaires (e.g. temps, longueur, dimensionnalité, etc.)
i, k, n, d	indices scalaires
t	indice temporel

\mathbf{x}	(gras) vecteur
\mathbf{x}^n	n ième exemple d'un vecteur d'état
$\bar{\mathbf{x}}$	valeur d'équilibre d'un vecteur d'état
\mathbf{x}^T	transposé d'un vecteur
$\hat{\mathbf{x}}$	estimateur d'un vecteur d'état
\mathbf{C}	(non italique gras) matrice
\mathbf{C}^{-1}	inverse d'une matrice
$\mathbf{u}^T \mathbf{u}$	produit scalaire

a, α, λ	quantité scalaire
a_i	i ième attribut d'un vecteur ou élément d'un ensemble
a_i^n	n ième exemple de a_i
x, y	coordonnées bidimensionnelles
θ	quantité angulaire
\hat{y}	estimateur d'une quantité scalaire
E	somme des erreurs au carré

A	(non italique) variable aléatoire
A_i	i ième variable aléatoire d'un ensemble
$Y = y$	réalisation d'une variable aléatoire Y
Y_t	variable aléatoire Y à l'instant t

$Y_{1:T}$	séquence temporelle ordonnée de T variables aléatoires Y_t
Q_t^d	d ième variable aléatoire Q à l'instant t
$Q_t^{1:D}$	ensemble ordonnée de D variables aléatoires Q à l'instant t
$P(\bullet)$	probabilité
$P(\bullet \bullet)$	probabilité conditionnelle
Δ	petit changement sur une quantité
$\ \bullet\ $	norme euclidienne
$\frac{\partial}{\partial x}$	dérivée partielle selon x
$[a,b]$	intervalle fermée sur a,b
$1:T$	séquence ordonnée de 1 à T
$\in \mathbb{R}^M$	élément de l'ensemble réel en M dimensions

AVANT-PROPOS

L'apprentissage machine, un aspect de l'intelligence artificielle, est une discipline où des algorithmes basés sur des méthodes statistiques sont utilisés de sorte qu'une machine (e.g. un processeur d'ordinateur) peut engendrer un processus de décision en développant ses facultés de généralisation à partir d'exemples.

Dans ce projet, on s'interroge sur la façon d'utiliser l'apprentissage machine, et plus particulièrement une forme spécifique de modèles graphiques probabilistes (PGM), les réseaux bayésiens (BN), pour analyser en temps réel les trajectoires bidimensionnelles d'agents mobiles.

Le contexte d'application de ce projet de recherche se situe dans le paradigme de *RoboCup* où des joueurs de soccer robotisés s'opposent au sein de diverses compétitions réelles ou simulées. *RoboCup* est une initiative internationale ayant pour objectif de promouvoir la recherche dans les domaines de l'intelligence artificielle et de la robotique. Le but ultime visé par *RoboCup* est de réussir à développer une équipe de soccer composées de robots humanoïdes pleinement autonomes capable de battre l'équipe humaine championne mondiale en 2050.

CHAPITRE 1 : « LES RÈGLES DU JEU »

Avant d'entreprendre le jeu, il convient de situer correctement la portée du projet, d'en comprendre l'intérêt et les divers aspects tels que les objectifs, le domaine d'application, la plateforme de travail, et de clarifier quelques concepts de base nécessaires à l'édification du modèle.

1.1 Mise en situation

Les Canadiens de Montréal jouent pour leurs fidèles partisans; c'est le temps de la traditionnelle soirée du hockey. Les spectateurs observent durant trois périodes de jeu de 20 minutes les faits et gestes des joueurs. Tout le monde, consciemment ou inconsciemment, y va de ses propres analyses, de ses déductions et prévisions plus valables que celles des autres. Celles et ceux qui écoutent la partie à la télévision laissent probablement Pierre et Yvon, les commentateurs de RDS, faire le travail de description du jeu. Il ne s'agit pas là d'une description simpliste et totalement objective des actions des joueurs comme par exemple, « avance d'un pas à gauche ... puis de trois autres devant ... se retourne ... élance le bras, etc. », mais plutôt d'un amalgame de descriptions, de prédictions et d'analyses, simples ou complexes, le tout relevé de quelques statistiques et d'une foule d'émotions qui trahissent généralement la véritable pensée des commentateurs. Par exemple, un crescendo de la voix va généralement laisser transparaître la nature peu commune voire bizarre, incroyable, exceptionnelle ou fortuite d'une certaine séquence de jeu.

Cette capacité à décrire et à analyser le jeu, communément appelée « vision du jeu », est essentielle autant pour le spectateur, le commentateur, l'analyste que pour le joueur afin d'avoir une compréhension cohérente du fil des événements. En d'autres termes, un néophyte ne s'excitera probablement pas lors d'une échappée d'un joueur ou lors d'un deux contre

un et c'est précisément ce qui en fait un néophyte : son incapacité à voir le jeu pour ce qu'il est vraiment.

Dans le jargon du milieu, les amateurs s'entendent pour dire qu'un joueur avec une bonne vision du jeu est généralement un bon constructeur de jeu. Cette affirmation pourrait être reformulée avec des termes propres aux domaines de l'apprentissage machine, de la planification et du contrôle de la façon suivante : des modèles de perception et d'analyse décisionnelle de qualité sont les bases nécessaires à une planification efficace. Sans en faire un postulat, il est possible d'avancer avec conviction que cette prémisse est assez solide pour en faire le fondement derrière ce projet de recherche.

1.2 Où se situe l'intérêt ?

Le contexte scientifique d'application se situe quelque part à la croisée des domaines de recherche sur la localisation et la navigation en temps réel, la planification d'activité, le mappage de l'environnement et les processus d'exploration en robotique, le contrôle de robots autonomes et les systèmes multi-agent.

Plus spécifiquement, le centre d'intérêt concerne l'analyse qui peut être effectuée à partir des déplacements d'un agent mobile autonome, et éventuellement des déplacements de plusieurs agents. Un modèle d'analyse adéquat et suffisamment générique pour s'appliquer à une foule de modèles de « perception » (au sens de localisation) permettrait, dans un premier temps, de caractériser complètement les déplacements d'un agent (*e.g.* un opposant), mais aussi, du point de vue des modèles génératifs, de révéler une source riche en prémisses pour d'éventuels modèles de planification.

En somme, il est question d'un modèle capable de décrire en toute généralité et à divers niveaux d'abstraction les activités d'un agent mobile et ce, à partir de l'observation objective de ses déplacements dans le plan ... un peu comme le ferait un commentateur sportif !

Bien que les objectifs de cette recherche soient directement liés aux thématiques de la navigation et de la planification et, d'un peu plus loin, aux autres domaines mentionnés précédemment, il n'est aucunement question de s'attarder à développer quelconque outil ou modèle de perception et/ou de localisation et/ou de planification.

Les résultats de ce travail de recherche pourraient toutefois s'inscrire dans le développement d'un cadre de travail polyvalent pouvant servir de base dans la plupart des domaines mentionnés.

D'un point de vue plus personnel, ce travail se positionne, primo, comme une initiation rigoureuse aux réseaux bayésiens, secundo, comme une étude dont la problématique est d'importance en intelligence artificielle et en robotique sans pour autant être trop pointue et spécialisée, tertio, comme un complément direct des activités de recherche que je mène dans une entreprise où l'on développe des dispositifs RFID de localisation en temps réel. Étant un éternel optimiste, je m'interroge à savoir si, avec toutes les techniques actuellement existantes, il est possible de développer quelque chose de relativement simple dans une optique d'intégration pouvant aspirer combler certains aspects ciblés de l'immense fossé qui existe entre les capacités de l'être humain et les capacités que l'on voudrait voir transmises à une entité robotisée.

1.3 Pourquoi limiter l'analyse aux trajectoires bidimensionnelles ?

Les trajectoires bidimensionnelles d'un agent mobile demeurent l'élément fondamental de ce travail de recherche puisqu'il est question de faire la description à divers niveaux d'abstraction des activités d'un agent mobile à partir de l'observation objective de ses déplacements dans le plan. En d'autres termes, les activités plus complexes sont inférées à partir d'autres plus simples et ainsi de suite jusqu'à l'activité dans sa forme la plus simple,

c'est-à-dire sous forme de trajectoires bidimensionnelles. Il s'agit là d'une approche ascendante par le bas (*bottom-up*) qui s'avère fort pertinente dans l'optique où la source de données est inconnue. Aucune présupposition n'est faite de sorte que le modèle demeure aussi générique et largement applicable que possible.

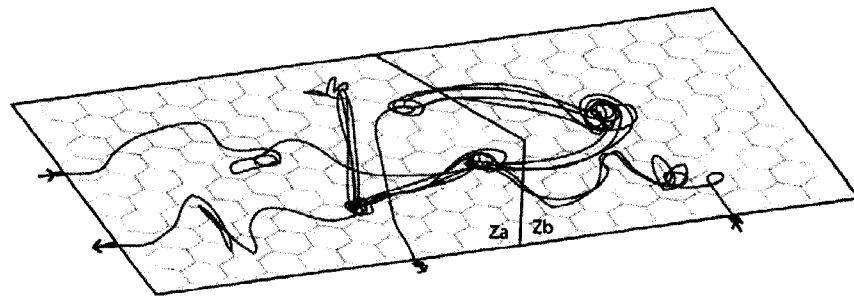


Figure 1.1 : Représentation dans le plan de deux trajectoires bidimensionnelles

Cela a toutefois pour conséquence de limiter la description des activités en terme de dynamique de la trajectoire, de patrons de déplacement et de localisation dans le plan. Toutes autres formes d'activité relative à un comportement physiologique plus complexe, notamment dans l'espace tridimensionnel (*e.g.* s'asseoir, sauter, pirouetter, se cacher, manipuler quelque chose, *etc.*) ne sont aucunement considérées.

1.4 Qu'est-ce que le modèle d'activité?

Jusqu'à maintenant le concept d'activité a été brièvement introduit de par la nécessité de discuter des activités d'un agent mobile autonome. Il convient de définir davantage le concept de modèle d'activité. Pour ce faire, les concepts de but, d'action, d'état et de primitives sont aussi définis.

Définition 1.1

Le concept de modèle d'activité est ici défini comme le schème de tous les préceptes qui orientent le choix des actions effectuées par un agent mobile autonome et, par conséquent, l'état de ce même agent.

Cette définition somme toute assez générale, voire même abstraite, présente l'avantage de s'appliquer à un large éventail de modèles référés dans ce travail, notamment dans les domaines de la planification, de la localisation, de la capture de mouvement, de la reconnaissance de comportement, des processus d'exploration et du mappage de l'environnement en robotique et du contrôle de robots autonomes.

Plusieurs auteurs (Baker, Tenenbaum, & Saxe, 2006; G. Kaminka, Fidanboyu, Chang, & Veloso, 2003; Pynadath & Wellman, 1995; Sukthankar & Sycara, 2005; M. Yin, Chai, & Yang, 2004) introduisent des concepts dans une terminologie différente, mais souvent analogue, qui finissent par se recouper. Il est fréquent de voir les termes « état » (de l'agent), « état mental », « action », « capacité », « but », « désir », « préférence », « directive » (*policy*), « environnement », « croyance » (de l'agent, différent du concept de croyance ou *belief state* dans un modèle probabiliste), « comportement », « comportement basique », « trace d'observations », *etc.* Il s'avère que les concepts d'état, d'action, de but et d'environnement suffisent généralement pour tout exprimer, du moins à un niveau de langage dissocié de tout formalisme mathématique.

Définition 1.2

Le concept de but est ici défini comme l'objectif ou l'aboutissement d'un plan ou d'une stratégie.

Un but peut être tantôt abstrait, tantôt concret. Du point de vue des modèles génératifs, le niveau d'abstraction d'un but s'illustre par le fait qu'une variable de type « but » peut produire d'autres variables de type « but »; il s'agit là de sous objectifs ou de buts se situant à un niveau d'abstraction moindre que celui de la variable génératrice. Par exemple, un agent mobile autonome participant à une partie de soccer peut avoir le but (abstrait) de gagner la partie. Ce même but peut être la source d'autres buts moins abstraits comme par exemple, « dominer l'adversaire durant la première demie », qui à son tour génère d'autres buts, toujours à des niveaux d'abstraction moindres, comme « marquer un point », « passer la balle rapidement », « se positionner en un lieu précis » et ainsi de suite.

Il est pertinent de remarquer que le concept d'action est intimement relié au concept de but.

Définition 1.3
Le concept d'action est ici défini comme l'une des réalisations d'un agent mobile autonome nécessaire à la concrétisation ou à l'accomplissement d'un but.

Puisque l'accomplissement d'un but (concret) se révèle comme l'une des réalisations nécessaires à l'accomplissement d'un autre but mais à un niveau plus abstrait, les concepts d'action et de but s'avèrent analogues mais à des niveaux d'abstraction différents. Pour reprendre les mêmes exemples, il semble logique que l'action de passer la balle rapidement soit l'une des réalisations nécessaires à l'accomplissement du but « marquer un point ». Lorsque accompli, le but « marquer un point » devient une action relativement à un but plus abstrait, c'est-à-dire une des réalisations nécessaires à l'accomplissement du but « dominer l'adversaire durant la première demie ».

La même réflexion peut s'appliquer au concept d'état de l'agent mobile.

Définition 1.4

Le concept d'état de l'agent mobile est ici défini comme la description objective à un instant donné des accomplissements de l'agent à un niveau d'abstraction plus élevé que celui de l'observation fondamentale (position bidimensionnelle).

Définition 1.5

Le concept de primitive est ici défini comme une portion de trajectoire indivisible (composante atomique) pour laquelle tous les points sont tirés d'une même distribution gaussienne.

Dans sa forme la plus simple, l'état de l'agent mobile à un instant donné correspond à la primitive observée caractérisant la trajectoire bidimensionnelle de l'agent à cet instant. Toutefois, à l'instar de la réflexion faite relativement au concept d'action, la description de l'état fondamental « se déplace selon une primitive donnée » peut devenir à un niveau plus abstrait « se déplace de telle façon (plus complexe) pour marquer un point ».

Afin de clarifier davantage l'étroite relation qui existe entre les variables de but, d'action et d'état, d'autres considérations plus formelles sont présentées à la section 3.2.

En définitive, il est bon de rappeler que tous ces concepts s'intègrent au sein du modèle d'activité et c'est précisément ce modèle d'activité, pour un agent mobile autonome spécifique, qui oriente la réalisation d'une quelconque suite d'actions. Le terme « comportement » est aussi fréquemment utilisé dans les contextes qui s'y prêtent pour évoquer ladite suite d'actions.

1.5 Quelques considérations

Les domaines d'application où il existe des systèmes de collecte de données provenant de trajectoires bidimensionnelles sont nombreux. Les systèmes de surveillance par vidéo, les systèmes de positionnement par réseau cellulaire, les réseaux de senseurs sans-fil et les systèmes de géolocalisation par satellite (GPS) figurent parmi les exemples. À cela peuvent s'ajouter une multitude de systèmes utilisés dans les sphères de la recherche mentionnées précédemment, parmi lesquels on trouve *RoboCup*, un paradigme de travail largement accessible intégrant une foule de défis technologiques liés de près ou de loin à la robotique.

Dans la plupart de ces domaines d'application, les questions suivantes sont d'intérêt :

- Quel est le but sous-jacent aux déplacements observés?
- Quelles sont les habitudes de déplacement de l'agent mobile considéré?
- Quelle est la destination la plus vraisemblable dans le futur?
- Est-ce que la trajectoire observée est anormale?

Avant même d'espérer pouvoir répondre à ces questions, il convient d'aborder de façon sommaire les principaux points susceptibles d'influencer la progression et les résultats de ce travail de recherche.

1.5.1 Les interactions avec les autres agents

Les interactions avec les autres agents mobiles sont d'une importance considérable et d'une complexité tout aussi grande. Les propos tenus précédemment quant à une vision du jeu de qualité, autant pour le joueur que pour le commentateur, sous-entendaient que les interactions entre les joueurs étaient nécessairement considérées. Cependant, cela ne va pas nécessairement de soi lorsqu'il s'agit d'intégrer ces interactions dans un même modèle d'analyse. En effet, l'évolution de la trajectoire d'un joueur

n'est plus uniquement dépendante du modèle d'activité de cet agent, mais aussi des états et des modèles d'activité de tous les autres joueurs. Les modèles considérant les interactions entre les joueurs sont souvent basés sur des structures organisées (e.g. arborescence) de scénarios prédéfinis, sur des patrons d'organisation spatiale prédéfinis, ou sur une abstraction de l'équipe entière comme une seule entité complexe (G. Kaminka, Fidanboyly, Chang, & Veloso, 2003; Kang, Hwang, & Li, 2006; P. Riley & Veloso, 2002; Sukthankar & Sycara, 2006).

Jusqu'à ce jour, peu de modèles intègrent, à la manière de Saria & Mahadevan (2004), les interactions entre agents mobiles et les fonctionnalités d'analyse d'activité et/ou de déplacement et/ou de comportement en terme de dynamique de la trajectoire. L'intégration des interactions entre les agents mobiles ne figure pas parmi les objectifs de ce travail de recherche.

1.5.2 L'environnement

Le concept d'environnement ou de contexte regroupe généralement toute l'information extérieure aux activités de l'agent mobile (et des interactions avec les autres agents mobiles). Cette information est fort pertinente du fait que deux trajectoires identiques ou similaires issues des activités d'un même agent mobile peuvent évoluer de façon différente selon l'environnement dans lequel se trouve l'agent.

En ce qui concerne les modèles graphiques probabilistes, tels que les réseaux bayésiens, la variable de type « environnement » est souvent une variable racine qui figure comme le parent de la plupart des autres variables (Baker, Tenenbaum, & Saxe, 2006; Pynadath & Wellman, 1995). Cette façon de faire accroît l'espace des états de façon exponentielle avec le nombre d'états que peut prendre ladite variable.

De plus, des repères spatiaux sont parfois incorporés dans la variable de type « environnement » afin d'influencer les variables d'action et d'état en fonction du lieu où se trouve l'agent mobile. Cette façon de faire est en partie compensée par les variables de type « but ». Il peut toutefois être utile d'ajouter de l'information complémentaire sur le repère spatial comme par exemple la configuration des lieux (Sukthankar & Sycara, 2005; , 2006).

1.5.3 Les buts

Le concept de but a déjà été introduit dans l'optique d'une intégration au sein du modèle d'activité d'un agent mobile autonome. Certaines considérations ont toutefois été passées sous silence. Par exemple, Baker, Tenenbaum & Saxe (2006) considèrent la variable de type « but » comme une variable statique analogue à la variable de type « environnement ». Toutefois, il est légitime de penser que les buts d'un agent mobile sont sujets à changer au fil du temps. Il est aussi légitime de penser qu'un agent mobile peut envisager plusieurs buts simultanément, que les buts puissent être structurés et priorisés. Dans un cadre multi-agent, il est possible qu'un même but soit partagé par plusieurs agents mobiles, comme c'est le cas par exemple pour une équipe de soccer par rapport aux buts « marquer un point » et « gagner la partie ».

Dans la majorité des cas, le dessein de la tâche d'apprentissage, c'est-à-dire la nature d'un but étant donnée, une certaine trajectoire bidimensionnelle, n'est même pas connue d'avance de sorte que l'apprentissage supervisé est souvent impraticable. Par exemple, dans le contexte d'une partie de soccer, un joueur a beau répéter certains déplacements particuliers, comment savoir à quels moments précis un certain type de jeu/manœuvre/but finit ou commence? Puisque cette information est souvent inconnue, l'apprentissage du modèle doit être effectué dans un contexte non supervisé et, par conséquent, le modèle ne peut que décrire objectivement les buts en terme

de déplacements, de dynamique de la trajectoire et de patrons de primitives.

1.5.4 Les périodes récurrentes

La récurrence des périodes fait référence à une conception de la ligne du temps en terme de périodes cycliques plutôt qu'en terme de temps absolu comme, par exemple, « semaine », « journée », « nuit », « avant-midi », « lunch », « heure de la pause », ou encore, dans une optique plus sportive, il pourrait s'agir d'une partie, « saison régulière », « éliminatoire », « du samedi soir », « en première demie », « en deuxième demie », « en prolongation », *etc.* Cette information peut être intégrée à l'information sur l'environnement ou encore être dissociée afin de mettre en relief la récurrence de certains patrons en fonction de ces périodes. Toutefois, l'intégration de cette information au modèle ne figure pas parmi les objectifs de ce travail de recherche étant donné le haut niveau de sophistication d'une telle variable.

1.5.5 Les anomalies

La détection des anomalies dans une trajectoire cache une problématique qui n'est pas nécessairement triviale : les patrons fréquents ne sont pas nécessairement les plus intéressants. En d'autres termes, la détection d'anomalies est directement liée à la recherche de points atypiques (*outliers*). Il est prévu de développer un modèle capable d'apprendre malgré l'existence d'anomalies dans les exemples. La détection d'anomalies est toutefois un objectif qui pourra être abordé ultérieurement sur les bases du présent travail. Liao, Fox & Kautz (2004) proposent d'ailleurs une approche en-ligne basée sur la sélection de modèles qui pourrait s'appliquer au modèle d'activité considéré dans ce travail.

1.5.6 La résolution

En considérant qu'il est toujours possible de standardiser les formats de données de type trajectoire provenant de diverses sources de données, les principales causes de discordance entre les sources se résument aux différentes résolutions spatiale et temporelle. Finalement, il est normal que la résolution de la fenêtre temporelle au travers de laquelle les données sont analysées, tienne compte des deux premières quantités.

A. La résolution spatiale

La résolution spatiale s'apparente à la précision spatiale des données constituant la trajectoire bidimensionnelle. D'emblée, il est possible que ces données soient continues, telles des coordonnées bidimensionnelles, ou discrètes, telles les cellules d'une grille. Dans le cas de données discrètes, il est possible que les cellules soient régulières ou non. Ces divers aspects de la résolution spatiale sont abordés brièvement à la section 4.1.

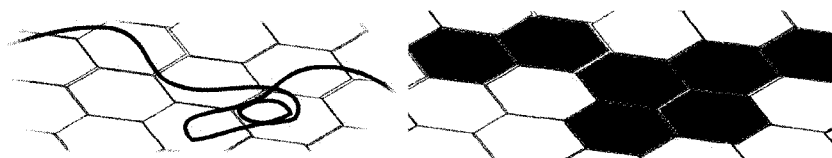


Figure 1.2 : Représentations d'une même trajectoire à deux résolutions spatiales distinctes. À gauche, les données bidimensionnelles sont continues. À droite, les données bidimensionnelles sont discrétisées en cellules régulières.

Dans le cadre de ce travail, il est entendu que la résolution spatiale reste invariante dans le temps.

B. La résolution temporelle

La résolution temporelle s'apparente à la fréquence d'échantillonnage des données constituant la trajectoire bidimensionnelle. Cette quantité est directement liée à la dynamique de la trajectoire et affecte nécessairement le processus d'analyse effectué par le modèle d'activité.

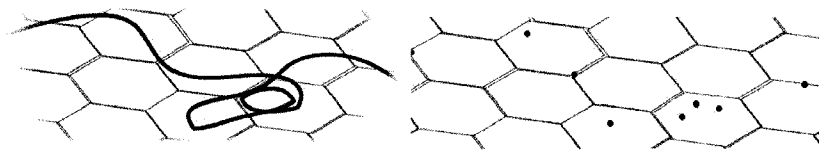


Figure 1.3 : Représentations d'une même trajectoire à deux résolutions temporelles distinctes. En gauche, les données bidimensionnelles sont échantillonnées à une fréquence nettement plus élevée qu'à droite.

En effet, il suffit de considérer dans un premier temps un agent mobile pour lequel trois données bidimensionnelles sont reçues à intervalle d'une minute. La première donnée correspond à un point A, la seconde, à un point B alors que la dernière correspond à un point C.

En considérant maintenant le même agent mobile pour lequel les trois mêmes données bidimensionnelles (points A, B et C) sont reçues à intervalle d'une seconde, il sera impossible pour le modèle d'activité de distinguer les deux trajectoires si la résolution temporelle n'est pas aussi considérée par le modèle.

En réalité, le fait que la première trajectoire se soit déroulée en deux minutes alors que la seconde, bien qu'identique, se soit déroulée en deux secondes change complètement la dynamique de la trajectoire et, par conséquent, l'analyse qui en résulte.

D'autres considérations quant à la fréquence d'échantillonnage des données bidimensionnelles sont abordées à la section 4.1.1.

C. La résolution de la fenêtre temporelle

La résolution de la fenêtre temporelle correspond à la largeur de fenêtre utilisée dans le modèle d'activité en terme de nombre d'échantillons (ou d'un multiple de la fréquence d'échantillonnage mentionnée au paragraphe précédent). Cette quantité est essentielle pour la reconnaissance des primitives et des patrons d'activité aux divers niveaux d'abstraction.

1.5.7 Le traitement en temps réel

Une considération d'importance pour le modèle d'activité est sa capacité de traitement en-ligne et idéalement en temps réel. En effet, dans l'optique de la mise en situation présentée en début de chapitre, un quelconque modèle d'analyse a bien peu d'utilité quant à l'efficacité d'une éventuelle planification s'il ne peut guère être utilisé en-ligne. Une bonne vision du jeu a peu d'importance si on a tout le temps désiré pour l'analyser...

Bien que le traitement, c'est-à-dire tout le processus d'inférence du modèle d'activité, doive être effectué en-ligne, il n'est toutefois pas nécessaire que la phase d'apprentissage le soit. Un apprentissage en-ligne sera toujours un avantage intéressant, par exemple, lors de compétitions *RoboCup* où l'adversaire n'a jamais été rencontré précédemment, mais il est fort acceptable qu'un modèle soit entraîné uniquement entre ses exécutions. C'est d'ailleurs dans ce contexte d'entraînement que le modèle d'activité considéré est utilisé.

1.5.8 Le bruit

Les trajectoires bidimensionnelles considérées sont bruitées de sorte que le modèle d'activité n'en devient que plus robuste. Tel qu'explicité au chapitre 4, les opérations de prétraitement des données bidimensionnelles sont toujours effectuées dans l'optique d'obtenir des primitives issues de distributions gaussiennes multivariées de sorte que le bruit considéré est lui

aussi gaussien. Tous les générateurs employés considèrent un bruit gaussien.

1.5.9 Les considérations retenues

En définitive, les considérations retenues quant au travail présenté sont résumées ci-dessous :

- Modèle d'activité pour un agent mobile unique, *i.e.* indépendant des interactions avec les autres agents
- Modèle d'activité indépendant de l'environnement
- Modèle d'activité intégrant les buts comme un ou plusieurs processus dynamique(s) structuré(s) et priorisé(s)
- Modèle d'activité indépendant des périodes récurrentes
- Modèle sans détection d'anomalies
- Modèle pouvant s'appliquer à diverses résolutions spatiales invariantes dans le temps
- Tests se limitant aux données bidimensionnelles continues
- Modèle pouvant s'appliquer à diverses résolutions temporelles
- Modèle d'activité pouvant s'appliquer à diverses résolutions de fenêtre temporelle
- Traitement en-ligne et en temps réel
- Observations affectées d'un bruit gaussien
- Apprentissage non supervisé

1.6 Pourquoi les modèles graphiques probabilistes ?

Il convient d'abord de définir correctement les concepts de modèle graphique probabiliste, de graphe orienté acyclique et de réseau bayésien. Les principaux concepts théoriques sont présentés à la section 4.3. Plusieurs travaux de recherche basés sur ce type de modèles sont mentionnés au prochain chapitre.

Définition 1.6

Un modèle graphique probabiliste (PGM) est un graphe où chaque nœud représente une variable aléatoire et les arêtes manquantes représentent les relations d'indépendance conditionnelle entre les variables déconnectées.

Définition 1.7

Un graphe orienté acyclique (DAG) est un graphe ne contenant pas de cycle orienté, c'est-à-dire que pour chaque vertex constituant le graphe, il est impossible de suivre un chemin orienté qui revient vers ce même vertex.

Le fait qu'un PGM soit un DAG signifie qu'il n'existe aucune relation causale cyclique donc aucune explication circulaire.

Définition 1.8

Un réseau bayésien (BN) est un PGM respectant les conditions qui définissent un DAG.

Dans un réseau bayésien, les arêtes sont orientées dans le sens de la relation de causalité (directe) entre deux variables aléatoires. La variable à la pointe d'une arête est appelée « enfant » alors que la variable à l'origine de l'arête est le « parent ». L'ensemble des variables « parent » pointant vers une variable aléatoire X_i donnée est noté $parents(X_i)$. Ainsi, un réseau bayésien est une représentation compacte de la distribution de probabilité jointe d'un ensemble de variables $\{X_1, \dots, X_n\}$ qui peut être calculée à partir du produit des distributions de probabilité conditionnelle locale entre une variable et ses parents.

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | parents(X_i)) \quad (1.1)$$

Maintenant, pourquoi s'intéresser aux PGM?

Les PGM présentent l'avantage de faciliter l'intégration de nouvelles variables de par la simple connaissance des liens de dépendance, ou plutôt d'indépendance, conditionnelle entre les variables. Cet avantage est indéniable dans l'optique de l'élaboration d'un modèle graphique probabiliste quelconque à partir de facteurs/variables encore inconnus. Les aspects probabiliste et génératif s'avèrent aussi des atouts incontournables, non seulement en ce qui a trait au modèle graphique lui-même, mais aussi par rapport aux modèles PCA et PPCA utilisés dans le prétraitement des données bidimensionnelles.

En quoi l'approche générative offre-t-elle un avantage sur l'approche discriminative, probabiliste ou déterministe?

Tout d'abord, il pertinent de mentionner que l'approche générative est nécessairement probabiliste. Le tout peut s'illustrer à partir de la règle de Bayes appliquée à un système ayant une sortie scalaire y et une entrée $\mathbf{x} \in \mathbb{R}^M$,

$$P(Y = y|\mathbf{x}) \propto P(\mathbf{x}|y)P(Y = y) \quad (1.2)$$

Le terme $P(\mathbf{x}|y)$ est un exemple de modèle génératif puisqu'il est possible de générer de nouveaux exemples de \mathbf{x} étant donné la sortie y . La règle de Bayes permet d'inverser le modèle et d'inférer la valeur en sortie en fonction de celle en entrée. Il s'agit alors d'une approche discriminative probabiliste, où l'on cherche à apprendre la postérieure $P(Y|\mathbf{x})$, souvent plus simple à modéliser que la jointe, afin de discriminer la classe de sortie à partir de la probabilité la plus élevée. Un exemple de ceci serait un réseau de neurones avec une fonction logistique (ou *softmax*) en sortie.

Dans une approche discriminative déterministe, on va chercher à estimer directement y par \hat{y} sans même passer par l'évaluation d'une distribution

de probabilité. Un exemple de ceci serait un réseau de neurones avec des sorties de type exclusif (*winner-takes-all*), ou encore, un modèle non paramétrique des plus proches voisins.

En somme, une approche discriminative modélise la frontière de décision plutôt que la distribution de probabilité complète derrière la réalisation de chacune des variables aléatoires. L'apprentissage supervisé est applicable aux deux approches puisque dans un cadre supervisé, il est possible pour un modèle discriminatif d'apprendre la sortie à partir des entrées sans pour autant connaître le processus génératif sous-jacent. L'approche générative devient toutefois une nécessité lorsqu'il est question d'apprentissage non supervisé et même semi supervisé.

Une fois entraîné, un modèle génératif permet de s'adonner au raisonnement prédictif en observant une variable et en essayant d'en inférer les conséquences possibles, ou encore, au raisonnement déductif en observant une variable et en essayant d'en trouver les causes possibles. La jointe étant connue, il est toujours possible de calculer un estimateur (e.g. *maximum a posteriori*) sur n'importe quelle variable étant donné une certaine forme d'évidence sur d'autres variables. Dans le cas d'une séquence temporelle, le BN devient un DBN à partir duquel il est possible de calculer l'explication la plus probable derrière la réalisation de ladite séquence d'observations.

Ces avantages sont d'une pertinence cruciale dans le cadre de ce travail puisque le modèle d'activité considéré ici se veut, en premier lieu, un modèle descriptif d'analyse de trajectoires, en second lieu, un modèle prédictif quant aux activités d'un agent mobile autonome et, finalement, un éventuel modèle (génératif) de planification.

En d'autres termes, selon l'utilisation qui en est faite, le modèle d'activité permet de générer des séquences d'actions et d'états à partir de préceptes

connus, ou encore, d'analyser les causes telles des préceptes abstraits et cachés à partir d'une suite donnée d'actions et d'états.

1.7 La problématique et les objectifs

L'objectif principal de ce travail de recherche consiste à élaborer, dans le paradigme des modèles graphiques probabilistes et plus spécifiquement des réseaux bayésiens, un modèle d'analyse des trajectoires bidimensionnelles effectuées par un agent mobile autonome. Ce modèle d'analyse se voit conféré la double fonction déductive/descriptive/explicative et générative propre aux modèles génératifs. Dans cette optique plus générale, il convient de modifier légèrement la terminologie pour parler du *modèle d'activité d'un agent mobile* plutôt que du *modèle d'analyse des trajectoires d'un agent mobile*.

Le modèle d'activité est ici abordé indépendamment des interactions avec les autres agents mobiles, de l'environnement et des périodes récurrentes.

La problématique demeure toutefois assez complexe. Tout d'abord, l'un des principaux défis consiste à intégrer dans un même réseau bayésien les buts de l'agent mobile, potentiellement abstraits, et ses actions en terme de dynamique de la trajectoire. Cet aspect est d'autant plus complexe qu'il faut l'effectuer dans un cadre non supervisé uniquement à partir de l'observation objective des déplacements de l'agent dans le plan. Cela implique de déduire les buts en reconnaissant les habitudes de déplacement de l'agent mobile sans nécessairement savoir quels patrons sont les plus pertinents.

Aussi, le fait qu'un agent mobile puisse envisager simultanément plusieurs buts distincts à divers niveaux d'abstraction implique un traitement de la trajectoire au travers d'une fenêtre temporelle à diverses résolutions.

Le modèle d'activité doit, dans son ensemble, être suffisamment générique et réutilisable afin de s'appliquer à la plus grande variété possible de

systèmes. Cela signifie que la problématique des diverses résolutions spatiales et temporelles des divers systèmes doit être attaquée.

Puisque la variabilité des trajectoires n'a aucune limite, c'est-à-dire que le nombre de trajectoires différentes est infini, il convient d'introduire l'espace des primitives, un nouvel espace plus favorable et de plus grande dimensionnalité dans lequel le nombre d'entités est potentiellement fini. L'idée consiste ensuite à traiter les trajectoires bidimensionnelles comme des séquences de primitives. La problématique de la définition de l'espace des primitives est loin d'être triviale.

Finalement, le calcul d'inférence nécessaire au traitement du processus d'analyse doit s'effectuer en temps réel. Une approche en-ligne pour un PGM n'est pas nécessairement viable en temps réel. Puisque cet aspect est directement lié à la complexité du modèle d'activité, l'ajout d'un modèle de segmentation simple et efficace est à évaluer afin d'alléger la charge de calcul au niveau du PGM.

1.8 Une approche non supervisée

Puisque les préceptes du modèle d'activité varient selon le champ d'application, l'approche non supervisée présente l'avantage de ne pas limiter le modèle à des valeurs fixées d'avance. De plus, dans plusieurs cas, les préceptes à apprendre sont inconnus de sorte qu'un apprentissage supervisé s'avère impraticable.

En supposant que les préceptes soient connus d'avance, un apprentissage supervisé nécessiterait d'identifier chacune des primitives d'une trajectoire en terme de ces préceptes de même que le commencement et la fin de chaque primitive, tâche fastidieuse et pratiquement impossible à faire à moins de s'attarder à la construction d'une base de données contenant l'éventail des préceptes et des primitives possibles ainsi qu'une panoplie d'échantillons pour chacune des situations envisageables.

L'apprentissage est effectué avec l'algorithme de maximisation de l'espérance (EM). La fonction de perte est la log-vraisemblance qui doit être minimisée (ou plutôt la log-vraisemblance négative qui doit être maximisée). Afin d'éviter le sur-apprentissage, dans le paradigme bayésien, il est possible de spécifier des *a priori* sur les CPD. Dans le cas d'une distribution multinomiale (CPT utilisée pour les variables discrètes), la distribution conjuguée *a priori* est une distribution de Dirichlet. Dans le cas d'une distribution gaussienne, un *a priori* est fixé sur la matrice de covariance lors de la réestimation des paramètres (étape M de maximisation de EM).

1.9 Pourquoi RoboCup ?

Tout d'abord, il faut souligner que la paradigme *RoboCup* est quand même éloigné de ce qui est généralement entendu par la robotique industrielle conventionnelle. L'application de la robotique au soccer n'est qu'un prétexte pour mettre sur pied un projet d'intégration technologique unique, largement accessible, riche en documentation, susceptible d'intéresser une majorité d'individus (considérant que le soccer est un sport mondialement populaire) et pouvant faire rêver une même communauté de chercheurs pendant près de 50 ans (il s'agit d'un projet de longue haleine dont l'objectif est d'aboutir d'ici 2050).

En effet, on retrouve au sein des diverses compétitions *RoboCup* les avancées les plus récentes sur les systèmes de vision, de localisation, de navigation et d'exploration, de communication sans fil, de contrôle, de planification, d'analyse de comportement et bien d'autres encore.

Dans le cadre de ce travail, les données provenant de diverses finales des compétitions *RoboCup Simulation League* entre 2003 et 2007 sont utilisées. Les fichiers de données de telles compétitions sont largement répandus sur Internet. Ces fichiers ont l'avantage de pouvoir être rejoués dans un lecteur disponible sur la plateforme de développement et exportés dans un format

ASCII contenant les positions bidimensionnelles du ballon ainsi que celles des 11 joueurs de chaque équipe (23 sources au total). La durée d'une partie est de 6000 cycles pour environ 10 minutes, soit environ 10 cycles / s

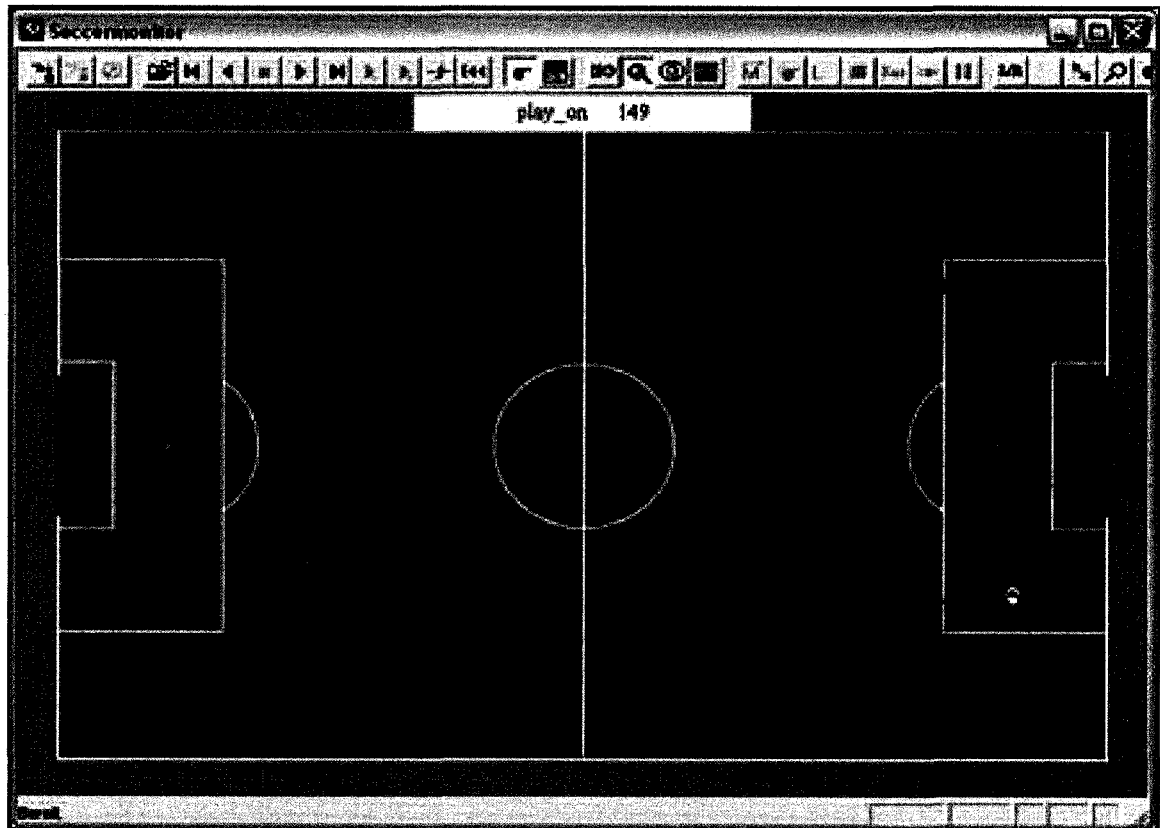


Figure 1.4 : Une partie dans le cadre de RoboCup Simulation League représentée à l'aide du module de relecture.

1.10 La plateforme de travail

La plateforme de développement du paradigme *RoboCup* n'est pas utilisée. Tous les développements et les tests sont effectués sur *Matlab* et sur *Simulink* étant donné qu'il est plutôt question de valider des algorithmes et des modèles mathématiques qui ne sont pas directement dédiés au cadre *RoboCup*. De plus, l'existence sur *Matlab* d'une librairie partagée sur les

réseaux bayésiens (Murphy, 2001) s'est avérée un facteur prépondérant quant au choix de la plateforme de travail.

CHAPITRE 2 : « L'AVANT-MATCH »

Un survol sommaire de la littérature est ici présenté afin de mettre en lumière les liens existant entre cette recherche et les divers contextes scientifiques d'application, notamment en planification et en analyse d'activité, en localisation et en navigation en temps réel, sur le mappage de l'environnement et sur les processus d'exploration en robotique, en contrôle de robots autonomes et sur les systèmes multi-agent.

Une emphase toute particulière est apportée aux techniques bayésiennes, notamment aux approches stochastiques markoviennes telles que le modèle markovien caché (HMM). Il est pertinent de remarquer que l'aspect hiérarchique revient fréquemment au sein des modèles qui sont maintenant l'état de l'art.

Les grands courants à l'origine des diverses approches dans les domaines de la planification et de l'analyse (reconnaissance, segmentation) de comportement et d'activité sont similaires en plusieurs points. En paraphrasant Plamondon & Shrihari (2000) dans leur revue de la littérature sur la reconnaissance d'écriture, aussi le résultat d'une certaine forme de trajectoires bidimensionnelles, mais dans les domaines d'intérêt ciblés par cette recherche, une première dichotomie serait celle des modèles « ascendants par le bas » (*bottom-up*) vs « descendants par le haut » (*top-down*), le premier étant généralement préféré dans les approches de bas niveau afin de construire à partir de prémisses simples (e.g. le modèle considéré dans ce travail), alors que le second est plus répandu au sein d'approches de haut niveau telles que les approches sémantiques et/ou symboliques (e.g. planification/reconnaissance de jeux dans RoboCup). Une autre dichotomie d'importance serait celle des « méthodes statistiques » vs « méthodes structurelles et par règles », le premier étant le paradigme principal de cette recherche parmi lequel se retrouvent les approches

discriminative et générative (e.g. HMM), alors que le second implique davantage une description abstraite de la séquence d'observations et l'utilisation de règles et de directives sur des quantificateurs et des prédicats.

Bien que les approches statistiques génératives « ascendantes par le bas » (*bottom-up*) soient plus fréquentes dans les travaux référés dans les prochaines sections, les approches alternatives « descendantes par le haut » (*top-down*) et/ou structurelles et par règles y figurent aussi et abordent souvent la problématique sous une perspective différente.

2.1 Point de vue de la planification

Dans une terminologie légèrement différente, Pynadath & Wellman (1995) mentionnent que la reconnaissance de plans se distingue de la reconnaissance de patrons par le fait que l'état de l'environnement influence l'état mental de l'agent mobile autonome, définissant ainsi un contexte pour la génération du plan. L'exécution du plan, c'est-à-dire des actions de l'agent mobile, affecte à son tour l'état de l'environnement. Baker, Tenenbaum & Saxe (2006) abordent la problématique de la planification dans le cadre des modèles génératifs comme étant l'inverse de la reconnaissance d'activité. Un modèle comportemental génératif est présenté qualitativement de même que la terminologie qui s'y rattache.

Le travail qui se veut le fer de lance de cette recherche est celui de Murphy & Paskin (2001), qui proposent un cadre générique et rigoureux pour un HMM hiérarchique (HHMM) au sujet duquel plus de détails sont donnés à la section 4.3.2. Le HHMM ne se veut pas explicitement limité à la planification. Il est mentionné que les grammaires probabilistes non contextuelles (*probabilistic context-free grammars*, PCFG) ont l'avantage d'offrir un modèle avec une hiérarchie d'une profondeur virtuellement sans limite alors que la topologie du DBN limite le HHMM à une profondeur fixée. Toutefois, l'algorithme d'inférence présenté pour le DBN est d'une complexité linéaire

avec la longueur de la séquence d'observations et rend le HHMM plus facile et efficace à entraîner.

Bui, Venkatesh & West (2001; , 2002) proposent un HMM abstrait (AHMM) appliqué à la reconnaissance de plan dans un édifice et reposant sur des concepts analogues au HHMM. Toutefois, à la différence de son homologue qui est développé dans le paradigme des DBN, le AHMM est développé à partir de processus markoviens de décision (MDP), ou plus spécifiquement, des processus markoviens de décision partiellement observables (POMDP). Un MDP représente une directive (*policy*), c'est-à-dire une séquence d'actions, à l'origine d'un processus stochastique markovien. Le POMDP est un MDP avec un modèle d'observation (Smith, 2007). Puisque la topologie du DBN final est légèrement plus spécifique que la topologie du HHMM, les concepts de buts, d'actions et d'états sont plus exclusifs les uns des autres. Bui (2003), Nguyen, Bui, Venkatesh & West (2003), Lühr, Bui, Venkatesh & West (2003), Bui, Phung & Venkatesh (2004) et Nguyen, Phung, Venkatesh & Bui (2005), sont toutes des extensions de AHMM et/ou HHMM qui abordent des problématiques telles que la dépendance non markovienne des états, la topologie du modèle et la performance des algorithmes d'inférence.

Le HHMM et le AHMM sont des modèles avec une indication sur la fin/terminaison d'une séquence d'états. Cette caractéristique s'avère cruciale lorsque plusieurs modèles (e.g. HMM) sont considérés simultanément. Il est pertinent de remarquer que plusieurs modèles présentés dans les sections qui suivent tentent par des heuristiques et/ou des sophistications d'intégrer des caractéristiques que l'on retrouve dans les HHMM et AHMM (e.g. structure hiérarchique, pluralité de modèles imbriqués, contrainte de terminaison, sélection du modèle le plus vraisemblable, etc.)

Liao, Fox et Kautz (2004) proposent une version AHMM sur mesure pour une application GPS sur les routines de transport des gens. Le modèle contient trois niveaux d'abstraction dans la hiérarchie et un apprentissage non

supervisé est effectué avec succès uniquement à partir d'observations sur la position et la vitesse. L'espace des états est toutefois connu et de dimension acceptable (*e.g.* marcher, autobus, auto, édifice, temps de début et de fin). Le modèle est appliqué à la détection d'anomalies.

Pynadath & Wellman (2000) ainsi que Pynadath (1999), privilégient une approche probabiliste s'inspirant du traitement du langage tel que les PCFG et les grammaires probabilistes dépendantes à un état (*probabilistic state-dependent grammars*, PSDG) par rapport aux PGM.

2.1.1 Les interactions multi-agent

Les travaux de Saria & Mahadevan (2004) étendent le modèle AHMM aux cas multi-agent en intégrant des directives dites « jointes » qui s'avèrent être des directives abstraites, donc au haut de la hiérarchie, et communes à tous les agents. Les auteurs diminuent aussi la complexité du processus d'inférence en introduisant des connaissances sur le début et la fin des directives (*policies*). Un aspect similaire est d'ailleurs discuté à la section 4.4. Un filtre à particule « rao-blackwellisé » permet de rendre les calculs d'inférence tractables. Toutefois, l'espace des états est similaire à ceux communément utilisés, c'est-à-dire que le niveau d'abstraction le moins élevé (le premier niveau dans leur modèle puisqu'il y a inversion de la nomenclature) correspond aux diverses cellules d'une grille bidimensionnelle et les niveaux plus abstraits correspondent à des buts du genre « tourner à gauche/droite » et « prendre la porte gauche/droite ». Finalement, puisqu'il s'agit d'un modèle de planification, les auteurs s'attardent aux activités générées par leur modèle et non à la capacité du modèle à décrire des activités à partir d'observations de bas niveau. En d'autres termes, les paramètres du modèle sont directement fixés à la main.

2.2 Point de vue des modèles comportementaux et d'activité

Hu, Xie & Tan (2004) proposent un modèle hiérarchique impliquant des réseaux neuronaux auto-organiseurs (HSOM) afin d'apprendre les patrons de trajectoires bidimensionnelles dans le cadre de la circulation automobile à une intersection et de déplacements piétonniers sur un campus. Le modèle est appliqué entre autres à la détection d'anomalies et à la prédiction de déplacements. Bien que très efficace, ce modèle s'applique à des positions bidimensionnelles dans l'espace absolu et est davantage adapté pour la caractérisation d'un environnement spécifique (*e.g.* l'intersection de deux rues) en terme de patrons de déplacement qu'à la caractérisation des déplacements d'un agent mobile particulier qui pourrait se trouver dans divers environnements.

Osentoski, Manfredi & Mahadevan (2004) utilisent un AHMM à deux niveaux pour classifier des trajectoires bidimensionnelles selon des patrons dans une optique analogue au modèle HSOM ci-dessus.

Yin, Chai & Yang (2004) proposent un modèle hybride DBN et N-gram pour la reconnaissance en-ligne d'activités de haut niveau à partir de données sensorielles bas niveau (puissance reçu à un point d'accès) dans un réseau sans fil (WLAN). Le DBN joue le rôle de modèle sensoriel permettant d'inférer des actions en terme de localisation dans l'espace et le N-gram tient le rôle de modèle d'activité permettant d'inférer des buts avec une technique de complexité linéaire selon l'ampleur de l'espace des états. L'entraînement est supervisé et les séquences d'observations et d'actions sont construites à la main puis associées à un but.

Yin, Shen, Yang & Li (2005) cherchent à inférer directement des états de haut niveau à partir de données de capteur de bas niveau sans avoir recours à une hiérarchie contenant une foule d'états et/ou d'actions intermédiaires difficile à apprendre dans un contexte non supervisé et difficile à obtenir dans un contexte supervisé. Le modèle proposé révèle quand même une

structure à deux niveaux, c'est-à-dire un HMM contenant un système linéaire dynamique (LDS) imbriqué. Le HMM décrit la dynamique non linéaire du processus stochastique impliquant des segments de bases provenant du LDS qui lui sert à capturer la dynamique locale des signaux de bas niveau. L'apprentissage des paramètres (HMM et LDS) du modèle probabiliste d'activité ainsi que des segments de bases et de leurs limites constituant le modèle de segmentation est effectué avec EM.

Li & Biswas (1999) proposent un modèle pour la reconnaissance de patrons comportementaux dans un contexte non supervisé basé sur la sélection de divers HMMs via une classification au travers d'un arbre binaire. Il s'agit là d'un bel exemple d'une approche générative hiérarchique montée sur mesure.

Sukthankar & Sycara (2005) proposent un modèle de reconnaissance de comportements humains appliqué aux opérations militaires en terrain urbain. Il ne s'agit donc pas de reconnaissance de trajectoires, mais plutôt de reconnaissance de mouvement. Toutefois, une certaine forme d'information spatiale est introduite dans la variable d'environnement puisqu'un agent peut se situer « sous un escalier », « derrière une cache », « dans la rue », *etc.* Une segmentation à temps fixe est appliquée puis deux poses subséquentes sont enchaînées sur lesquelles PCA est appliquée dans le but de classifier l'échantillon en une action physique avec une machine à vecteurs de support (SVM). La séquence d'actions physiques est filtrée par un HMM. Un comportement complexe correspond à une machine à états finis (FSM) dont les états correspondent à des actions physiques dans un environnement donné. La reconnaissance de la séquence de comportement s'effectue en choisissant la FSM la plus appropriée à partir d'une fonction de coût faite sur mesure. Il s'agit d'un autre exemple de modèle s'inscrivant dans les optiques génératives hiérarchiques monté sur mesure.

Ihler & Smyth (2006) proposent, quant à eux, un modèle bayésien non paramétrique apte à construire des profils d'intensité dans le temps au regard aux diverses activités humaines appliquées aux domaines tels que le RFID, la surveillance vidéo, les interactions humain-machine, l'analyse de données sur Internet, *etc.* Il est pertinent de remarquer que ce modèle répond à la problématique des périodes récurrentes énoncée à la section 1.5.4.

2.2.1 Le cas multi-agent

Sukthankar & Sycara (2006) présentent un modèle spatio-temporel de reconnaissance de comportements en équipe toujours appliqué aux opérations militaires en terrain urbain. Il s'agit de deux modèles basés sur la position bidimensionnelle de chaque agent et conçus pour se compléter. Le premier est un modèle structurel consistant en une série de patrons encodant des relations statiques entre les membres de l'équipe et des points de repère externes représentant des entités. La qualité de la correspondance entre les observations et les patrons est évaluée avec des transformations de similarité de rotation, de translation et de mise à l'échelle. Le second modèle est un HMM avec un espace des états encodant la position bidimensionnelle et la vitesse de tous les agents. Ce modèle est utilisé dans le cas spécifique des équipes en binôme où la première technique performe moins bien.

2.2.2 La reconnaissance visuelle

Robertson & Reid (2005) proposent un modèle hiérarchique sur mesure à deux niveaux, chaque niveau étant un HMM, pour faire la reconnaissance de comportements piétonniers et appliqués au tennis à partir d'observations vidéo. Les résultats obtenus sont éloquentes. Il est intéressant de noter que les observations brutes sont cumulées sous la forme de mini distributions similairement à ce qui est présenté à la section 4.2.

2.2.3 La navigation et le mappage de l'environnement

Roweis & Salakhudinov (2003) s'attaquent à la problématique de la localisation et du mappage sensoriel de l'environnement (*simultaneous localization and surveying*, SLAS¹) appliqué au cas multi-agent. Un agent est modélisé par un HMM avec un espace des états discrets (les cellules d'une grille) et, dans une autre version, avec un espace continu échantillonné avec un filtre à particules. Dans le cas multi-agent, le modèle devient un HMM factoriel avec une information de proximité jouant à titre de variable de jonction (*explain away*).

Kwok & Fox (2005) attaquent la problématique de l'interaction d'un agent mobile avec son environnement et proposent un modèle de localisation pour le suivi d'un agent mobile appliqué au robot AIBO dans le paradigme *RoboCup*. Un DBN arborescent (*tree-structured* DBN) intègre les états de l'agent suivi (le ballon dans le cas échéant) à deux niveaux d'abstraction (vecteurs position et vitesse en plus du mode de déplacement) et l'état de l'agent suiveur à partir des observations faites sur son environnement (des repères spatiaux).

2.3 Dans le cadre des compétitions de soccer multi-agent

S'inscrivant directement dans le contexte de cette recherche, Han & Veloso (1999) s'attardent à la reconnaissance en temps réel de comportements de joueurs de soccer robotisés dans la perspective où un agent observe la trajectoire bidimensionnelle d'un autre agent. La problématique de l'agent observateur/commentateur extérieur (*narrative agent*) est explicitement mentionnée. Étant donné les contraintes du problème, les auteurs se

¹ Le SLAS se distingue de la localisation et du mappage de l'environnement ou *simultaneous localization and mapping*, SLAM, par le fait qu'il ne s'agit pas uniquement d'évaluer l'encombrement de l'espace, mais plutôt d'une foule de signaux sensoriels tels que la température, la luminosité, etc.

penchent vers les modèles génératifs. Plus spécifiquement, ils modélisent chaque comportement de haut niveau avec un HMM doté d'états de quatre types: démarrage, intermédiaire, sortie et rejet. Une particularité du modèle concerne les caractéristiques observables qui ne sont pas les mêmes pour chaque HMM, donc variables selon le comportement qui est en train d'être évalué. Plus spécifiquement, un comportement d'approche du ballon considère un espace bidimensionnel compartimenté circulairement autour de la position courante du ballon alors qu'un comportement de dribble va plutôt considérer une compartimentation en quadrants et un vecteur vitesse. Afin de reconnaître un comportement, un nouvel HMM est instantié à période fixe et évalué tant qu'il n'atteint pas les états de rejet ou de sortie ou que le temps d'exécution maximal du comportement associé n'est pas écoulé. Le modèle est nécessairement limité par le nombre d'agents, le nombre de comportements alloués et le nombre d'états dans chaque HMM. Ceci est un autre exemple de multiples modèles montés sur mesure.

2.3.1 Le paradigme RoboCup

Riley & Veloso (2002) énoncent la problématique du « support à l'entraînement » (*coaching*) appliquée dans le cadre des compétitions *RoboCup*. L'entraîneur (*coach*) peut être vu comme une source d'information extérieure à l'environnement des joueurs et susceptible d'influencer leur comportement par des communications avec ces derniers. L'entraîneur est en quelque sorte un modèle d'analyse et de planification global. Riley (2005) nuance le concept à un agent susceptible de conseiller un autre agent sur la façon la plus appropriée d'agir dans un environnement donné. Riley & Veloso (2004) proposent un modèle de « support à l'entraînement » dans le cadre d'un MDP où l'espace des états est défini en terme de la position du ballon sur une grille, de but compté ou non, de possesseur du ballon et d'occupation du terrain.

Riley & Veloso (2002) développent un modèle probabiliste dans un cadre bayésien afin d'appliquer le concept d'entraîneur à la planification des activités de l'adversaire. Un classificateur bayésien naïf choisit le modèle probabiliste le plus probable parmi un ensemble de modèles représentant la défensive de l'adversaire telle une distribution de probabilité étant donné les positions observées des joueurs et du ballon.

Il existe beaucoup d'approches structurelles symboliques et/ou par règles qui s'avèrent souvent très spécifiques et difficilement transposables à d'autres champs d'applications étant donné la spécificité de certaines règles/prédicats. Kaminka, Fidanboyu, Chang & Veloso (2003) est un exemple d'approche structurelle par règles qui plus est descendante par le haut (top-down). Les auteurs partent du principe que les comportements atomiques (*e.g.* passe entre deux joueurs, interception par un joueur, dribble, *etc.*) peuvent être définis assez aisément par les experts du domaine et que la difficulté réside plutôt dans la combinaison séquentielle de ces comportements. Ils présentent une technique pour l'apprentissage non supervisé de comportements agent et multi-agent au sein d'une équipe sportive en traduisant la séquence des observations durant la partie en une séquence de comportements atomiques par équipe reconnus à partir de règles discrètes sur des quantificateurs et des prédicats. La séquence est ensuite compilée dans une structure en arbre sur laquelle des techniques de compte fréquentiel et de dépendance statistique sont appliquées pour valider l'importance relative de sous-séquences.

Lattner, Miene, Visser & Herzog (2006) présentent aussi une approche structurelle basée sur des règles complexes à partir de quantificateurs et des prédicats appliqués à des séries temporelles de positions, de distances, de vitesses et de directions.

Kaminka & Avrahami (2004) définissent un algorithme heuristique pour la reconnaissance de comportements basé sur une symbolique permettant de

faire correspondre des comportements à des observations, de propager des marqueurs de temps dans un DAG et d'extraire des hypothèses. Un arbre de décision est utilisé comme structure de contingence pour le couplage des observations à des comportements.

2.3.2 L'analyse de trajectoires

Vlachos, Kollos & Gunopulos (2002), Lee & Xu (2004) utilisent le concept de sous-séquence commune la plus longue (*longest common subsequence*, LCSS) pour classer des trajectoires.

Kang, Hwang & Li (2006) proposent un modèle d'analyse de trajectoires pour des joueurs de soccer dans le contexte du logiciel *Football Manager 2005* basé sur des régions d'intérêt qui sont calculées à partir de la trajectoire de déplacement. Le modèle permet d'intégrer les interactions entre les agents en liant les régions d'intérêt des divers joueurs afin de faire ressortir des zones neutres ou sécuritaires et des zones de conflit.

En définitive, il semble clair que les approches bayésiennes et plus spécifiquement les approches stochastiques markoviennes figurent comme l'une des pierres angulaires dans les domaines de recherche sur les modèles de planification et les modèles comportementaux et d'activité. L'aspect hiérarchique revient fréquemment parmi les modèles présentés et s'avère l'élément essentiel duquel ce travail est inspiré.

CHAPITRE 3 : « LA FORMATION DE L'ÉQUIPE »

Dans ce chapitre, une revue de la première approche tentée est sommairement présentée. Les limitations ayant menés à la seconde approche impliquant un modèle hiérarchique sont aussi mises en relief. Finalement, les avantages de l'approche hiérarchique appliquée aux objectifs de cette recherche sont énoncés.

3.1 La première approche considérée

Dans une première approche, des PGMs statiques (sans modélisation temporelle) ont été considérés pour construire le BN représenté à la Figure 3.1 qui peut s'interpréter comme suit : un agent mobile autonome ayant un but spécifique (*e.g.* une destination, inconnue du monde extérieur) prendra des actions spécifiques (*e.g.* un certain transit de déplacement, aussi inconnu du monde extérieur) qui auront pour effet de modifier son état (*e.g.* une position, aussi inconnue du monde extérieur) à partir duquel l'observation dont dispose le monde extérieur sera produite (*e.g.* un signal électromagnétique provenant d'un émetteur, un signal laser provenant d'un capteur optique, etc.). Il est pertinent de noter la convention suivante (utilisée tout au long de ce travail) : les nœuds foncés représentent des variables observées alors que les nœuds pâles représentent des variables cachées dont l'état est inconnu. Ce BN peut être décortiqué en deux BN plus simples représentés aux Figure 3.2 et Figure 3.3. Le BN de la Figure 3.2 se veut un modèle d'observation permettant d'inférer la chaîne « état, action, but » étant donné un contexte (l'environnement et la période récurrente) à partir d'une observation (*e.g.* la mesure d'un signal) faite sur la trajectoire bidimensionnelle d'un agent mobile. Le BN de la Figure 3.3 se veut un modèle de « genre prédictif » (car ce n'est pas de la prédiction à proprement parler) dans lequel l'action et le but de l'agent mobile sont déduits à partir d'une suite d'états passés et du contexte.

Une première limitation concerne les variables « action :transit » et « état(s) antérieur(s) » qui peuvent cacher un espace d'états passablement complexe si l'objectif est vraiment d'y intégrer des séquences temporelles. Premièrement, il faut nécessairement limiter la longueur de ces séquences temporelles. Deuxièmement, pour une séquence de durée fixée, le nombre de séquences distinctes est infini dans un espace continu, et exponentiel par rapport à la longueur de la séquence dans un espace discret fini. De plus, le modèle prédictif limite la dépendance du passé à la variable d'action alors qu'il est légitime de penser que la variable d'état peut aussi en bénéficier.

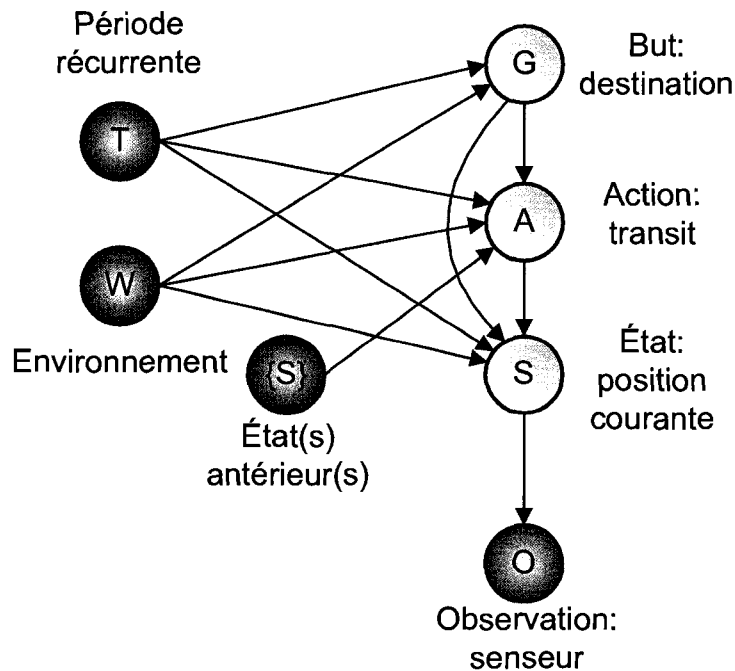


Figure 3.1 : Représentation du BN statique intégrant un modèle d'observation et de « genre prédictif » en un seul BN.

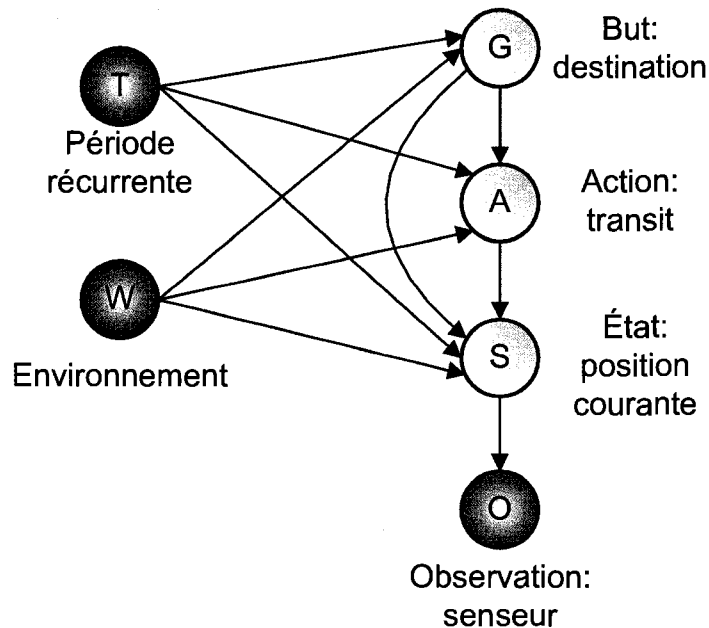


Figure 3.2 : BN représentant le modèle d'observation.

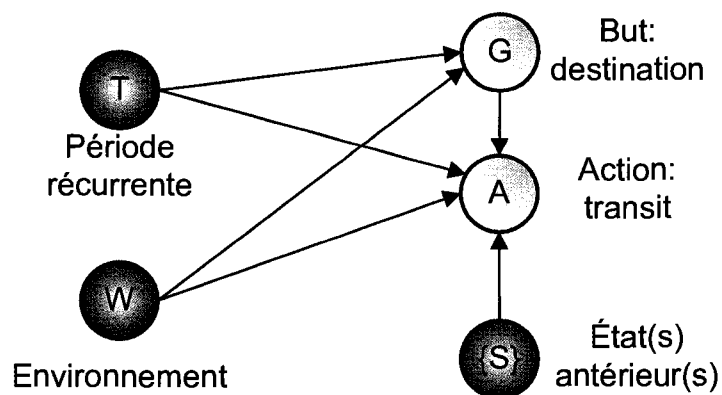


Figure 3.3 : BN représentant le modèle de « genre prédictif ».

En supposant que la longueur de la série temporelle soit limitée à une seule observation afin de réduire le nombre d'états possibles, le modèle n'attaque toujours pas les problématiques de la multiplicité des buts, de l'évolution temporelle des buts (et des actions) de même que celle des résolutions diverses.

3.2 L'alternative gagnante

Une seconde approche, plus générique, faisant intervenir un DBN s'avère beaucoup moins limitative. L'idée consiste à utiliser des HMMs imbriqués les uns dans les autres afin de former une hiérarchie telle que représentée à la Figure 3.4 et de produire des analyses similaires à celle de la Figure 3.5.

Puisque le modèle doit être en mesure de représenter la pluralité des trajectoires bidimensionnelles possibles, idéalement en évitant de discrétiser l'espace, le modèle d'observation est appliqué sur des primitives supposées gaussiennes et en nombre limité dans un espace de haute dimensionnalité.

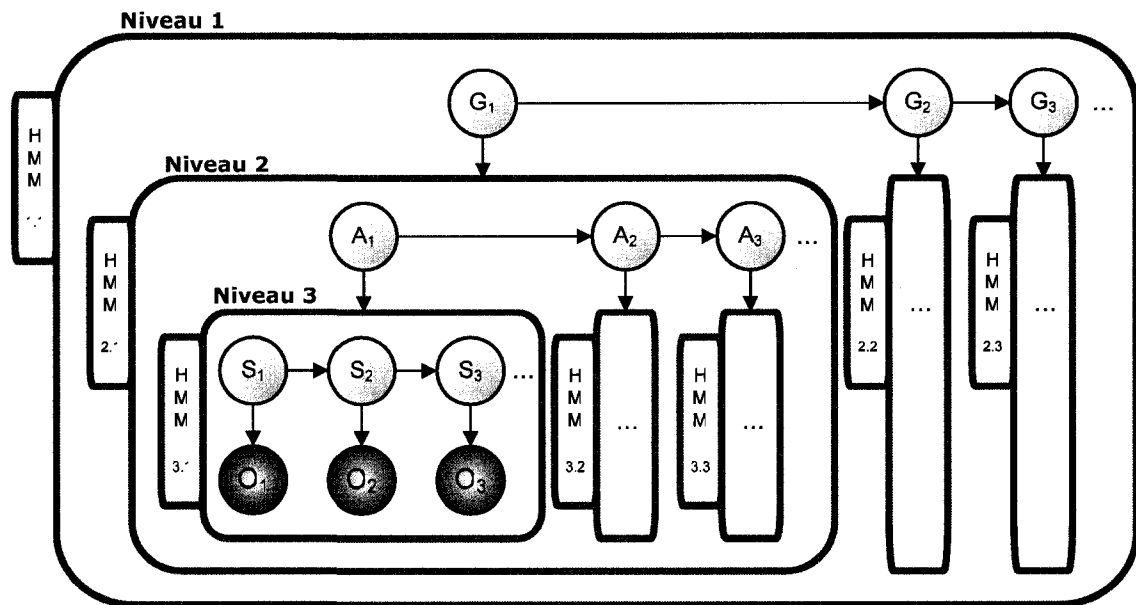


Figure 3.4 : Représentation schématique de plusieurs HMMs imbriqués les uns dans les autres afin de former une hiérarchie. Il s'agit ici d'une hiérarchie à trois niveaux où le niveau 1 représente le niveau parent (la racine de la hiérarchie) alors que le niveau 3 représente les feuilles de la hiérarchie (émission des observables). Un exemple générique d'une hiérarchie analogue (à quatre niveaux) est représenté à la Figure 3.7.

Les variables cachées représentent toutes le même concept de but/action/état (que l'on nommera simplement « but » par souci de simplicité) mais à divers niveaux d'abstraction. Ainsi la variable située à la racine de la hiérarchie représente le but le plus abstrait et celles produisant les observations représentent l'état le plus concret de l'agent mobile.

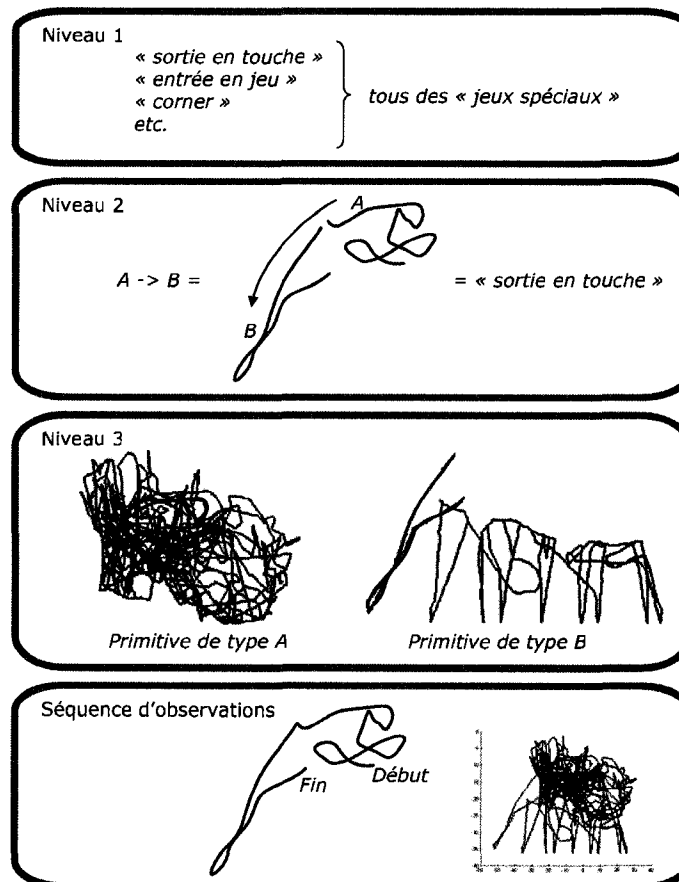


Figure 3.5 : Exemple du genre d'analyse qui pourrait être effectuée par un modèle hiérarchique comme celui de la Figure 3.4 sur une trajectoire bidimensionnelle provenant d'une partie RoboCup. Deux types de primitives pourraient être distinguables : le va-et-vient sur le terrain (A) et les déplacements directs liés à la bordure du terrain (B). Les sous-patterns A->B (« sortie en touche ») et B->A (« entrée en jeu ») seraient des réalisations possibles au niveau 2 d'un but plus abstrait « jeux spéciaux » au niveau 1.

Distinguer explicitement, c'est-à-dire avec des états différents, les concepts de but, d'action et d'état ne sont pas une absolue nécessité. Par exemple, une variable d'action pourrait encoder de l'information sur la dynamique de la trajectoire et une variable de but, encoder de l'information quant à des repères spatiaux. Cela aurait pour effet de limiter la hiérarchie (*e.g.* les niveaux 1 et 2 à la Figure 3.4) à des buts uniquement en terme de repères spatiaux de sorte qu'il deviendrait impossible d'intégrer l'information sur la dynamique de la trajectoire dans la hiérarchie. Pour cette raison, les concepts de but, d'action et d'état sont uniformisés. En d'autres termes, c'est comme si les variables *S*, *A* et *G* du BN à la Figure 3.3 étaient fusionnées en une seule mega-variable répliquée à divers niveau d'abstraction.

Dans cette nomenclature but/action/état (GAS), chaque élément peut être interprété comme un niveau (parmi trois subséquents) quelque part dans la hiérarchie. Dans ce cas, si l'on cherche à conserver les relations de dépendance telles qu'exprimées dans Baker, Tenenbaum & Saxe (2006) ainsi que Bui, Venkatesh & West (2002) sommairement représentées à Figure 3.6 (a)(b), alors la généralité du HHMM est perdue au profit d'un DBN avec une topologie sur mesure. Cependant, en considérant un concept unifié tel que représenté en (c), tous les niveaux de la hiérarchie conservent la même topologie qui est celle d'un HMM. Toutefois, il est vrai que des topologies comme celles de la Figure 3.6 (a)(b) ont l'avantage de contraindre la structure, ce qui pourrait s'avérer un avantage dans le cas d'un apprentissage non supervisé.

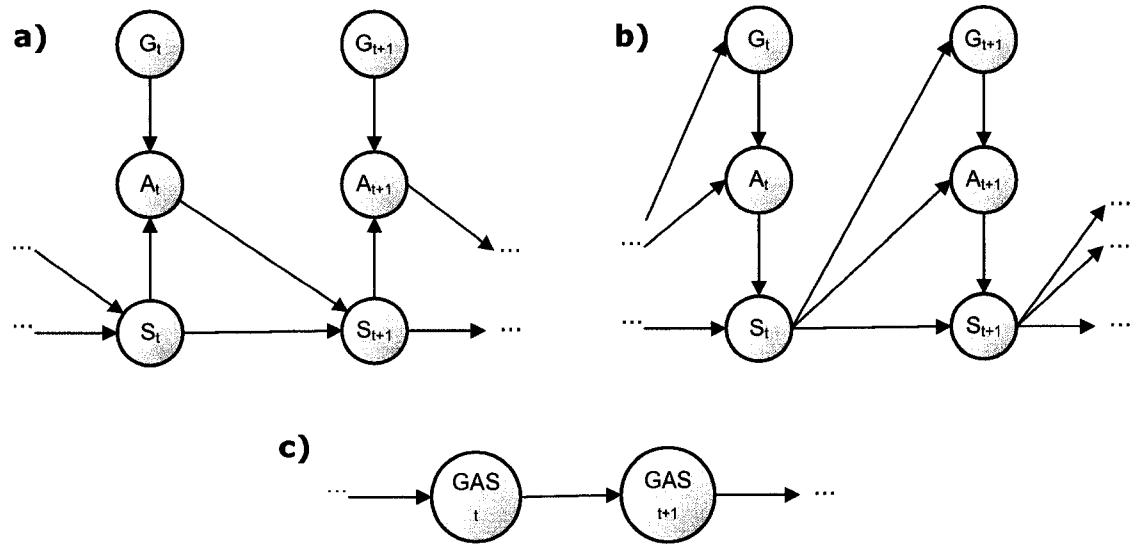


Figure 3.6 : En (a) le but et l'état influencent l'action qui influence à son tour l'état suivant, et de façon analogue, en (b), l'action influence l'état courant qui lui influence le but et l'action suivante. En (c), le but, l'action et l'état sont fusionnés dans une même variable qui, bien sûr, évolue vers son prochain état.

Dans l'optique d'une variable unifiée, la définition du concept d'état de l'agent mobile à un instant donné (Définition 1.4) peut s'interpréter, ultimement comme étant la réalisation de la variable au niveau produisant l'observation, mais aussi comme étant la réalisation de l'empilement complet de variables GAS à cet instant.

3.3 Les distinctions du modèle envisagé

La grande majorité des modèles d'activité hiérarchiques relatés au chapitre précédent considèrent des buts de haut niveau qui, au mieux, intègrent des repères spatiaux tels que « marcher vers le trottoir », « aller à la cuisinière », « sortir par la porte de droite », « se positionner derrière le ballon », etc.

Dans Bui, Venkatesh & West (2001), la hiérarchie de buts correspond à une classification hiérarchique physique réelle de points de repères dans l'espace. Par exemple, le premier niveau de la hiérarchie représente les portes d'un étage, le second niveau, les portes des différentes pièces de l'étage et le dernier niveau représente les positions discrètes sur une grille. Une telle hiérarchie, bien qu'applicable aux déplacements d'un agent mobile désirant « sortir » d'un étage, ne s'applique pas à un agent extérieur qui aurait pour objectif « d'entrer » sur l'étage.

Une première distinction réside donc dans le fait que le modèle d'activité proposé dans ce travail dissocie la structure hiérarchique du modèle de toute autre structure hiérarchique fixe. Plus généralement, le modèle d'activité se distingue par le fait qu'il considère (et se limite à) la dynamique de la trajectoire tout en s'attardant aux diverses considérations sur la résolution (spatiale, temporelle et fenêtre temporelle). Les variables de but du modèle d'activité intègrent via l'identité des patrons de primitives qu'ils décrivent, l'information sur la dynamique de la trajectoire, qualifiant ainsi les déplacements de l'agent mobile. Cela a cependant l'effet de limiter le niveau d'abstraction à des buts que certains qualifieraient de « plutôt concrets ». Des exemples de tels buts seraient « accélérer, s'arrêter et revenir sur ses pas », « se rendre directement à un point A » ou « effectuer le parcours P et s'arrêter dans les environs E ». C'est comme si l'apprentissage du modèle d'activité était limité à un certain niveau tel que cela est représenté schématiquement à la Figure 3.7. Il est pertinent de remarquer ici que, à la lumière des travaux présentés au chapitre précédent, les niveaux de hiérarchie plus abstraits peuvent aisément faire l'objet d'autres paradigmes potentiellement plus appropriés tels que l'apprentissage supervisé à l'aide d'experts, l'intégration d'interactions multi-agent, des approches de type descendant par le haut (*top-down*), etc.

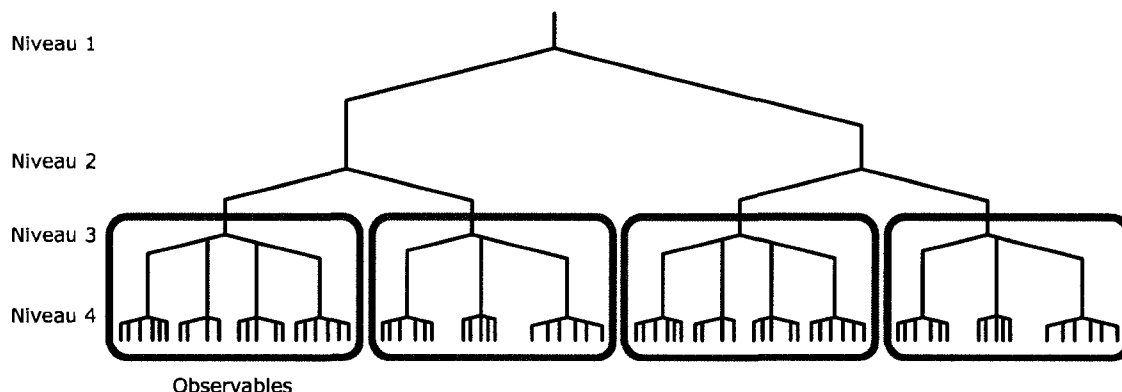


Figure 3.7 : Représentation schématic d'une hiérarchie de buts/actions/états. En coupant la hiérarchie à un certain niveau (ici au niveau 2), il en résulte une collection de hiérarchies plus simples à un niveau d'abstraction moindre. Par exemple, les quatre cas possibles du niveau 2 deviennent les racines de quatre hiérarchies plus simples aux niveaux 3 et 4. Ces hiérarchies peuvent servir à représenter des patrons de primitives dont le but ultime est l'un des quatre buts du niveau 2.

Ainsi, il existe dans le modèle considéré, pour une descendance donnée dans la hiérarchie, un lien direct entre le niveau d'abstraction et la résolution de la fenêtre temporelle appliquée sur la trajectoire bidimensionnelle. Étant donné qu'un but à un niveau d'abstraction donné est une composition de plusieurs buts à un niveau d'abstraction moindre, et considérant que l'accomplissement d'un but est un processus durant lequel s'écoule nécessairement un certain temps, alors le niveau d'abstraction d'un but est inversement proportionnel à la résolution de la fenêtre temporelle nécessaire pour caractériser la trajectoire.

Autrement dit, un but à la racine de la hiérarchie est plus abstrait étant donné sa « grande » fenêtre temporelle, et des buts au dernier niveau (production des observations) sont moins « abstraits » étant donné leur « petite » fenêtre temporelle. Par exemple, en supposant qu'une primitive ait une durée de 10 s et que la suivante ait une durée de 30 s, alors le but à

l'origine de ces deux primitives a nécessairement une fenêtre temporelle d'une résolution de l'ordre de 40 s voire même plus.

En somme, les motivations derrière le modèle hiérarchique ont été présentées de même que les distinctions de l'application qui en fait dans ce travail par rapport aux modèles présentés au chapitre 2. Avant de spécifier la méthodologie (chapitre 5) pour effectuer l'entraînement et les expérimentations (chapitres 6 et 7), il convient de faire un survol de la théorie concernant les HMMs et HHMMs.

CHAPITRE 4 : « L'ÉCHAUFFEMENT »

Il est question dans ce chapitre de présenter les bases théoriques de même que les diverses définitions et/ou modélisations nécessaires à l'édification du modèle d'activité. La modélisation multi-résolutions est d'abord présentée, puis suit la modélisation de l'espace des primitives en fonction des opérations de prétraitement. Les bases théoriques des HMMs et HHMMs sont sommairement exposées. Finalement, les modèles de segmentation en ligne considérés pour alléger la charge de calcul au niveau du PGM sont détaillés.

4.1 Modèle d'activité multi-résolution

La problématique de la résolution énoncée au chapitre 1 a pour objectif de s'affranchir de la technique d'observation. Pour ce faire, le traitement appliqué en entrée aux positions dans le plan doit être généralisé afin que les primitives utilisées au niveau du PGM puissent être construites sans égard à la nature des données fournies.

Tout d'abord, il faut remarquer ici qu'une position bidimensionnelle n'est pas nécessairement un point dans le plan. En effet, il pourrait aussi s'agir d'une zone ou d'une cellule particulière. L'idée derrière la généralisation des positions dans le plan est simple. Il s'agit d'identifier, à partir d'un identificateur unique, chaque position possible dans le plan.

Trois couches de représentation sont donc introduites :

- La couche (C1) continue de base sur laquelle l'identificateur unique d'une position ou d'un point de repère est représenté par les coordonnées $(x, y) \in \mathbb{R}$ de ladite position ou dudit point de repère.
- La couche (C2) discrète intermédiaire à une résolution fixée sur laquelle l'identificateur unique d'une cellule (*i.e.* une maille régulière) est représenté par les attributs $(xId, yId) \in$ quelconque ensemble réel. Ces

attributs identifient l'origine ou le centroïde de ladite cellule et pourraient correspondre, par exemple, à des coordonnées dans \mathbb{R} ou à l'index des colonnes et des rangées dans \mathbb{N} .

- La couche (C3) logique sur laquelle l'identificateur unique d'une zone logique de forme quelconque est représenté par un entier $zld \in \mathbb{N}$.

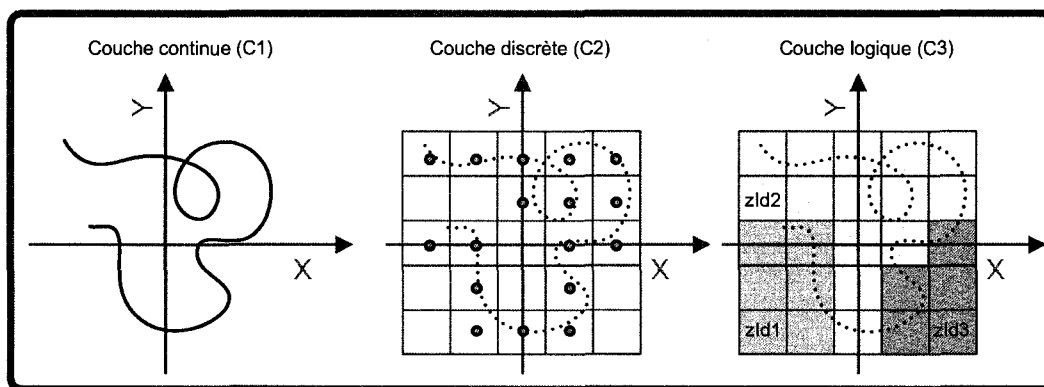


Figure 4.1 : Les trois couches de représentation permettent d'attaquer la problématique des diverses résolutions spatiales d'une position dans le plan.

À un instant donné, la position d'un agent mobile pourrait être traitée sur plus d'une couche afin de faire un traitement parallèle de la trajectoire à diverses résolutions spatiales. Toutefois, dans le cadre de ce travail, seule la couche continue (C1) est considérée. Par conséquent, le terme « donnée bidimensionnelle » sera employé dans le reste du document. Les autres couches de représentation feront l'objet de travaux ultérieurs afin d'être validées.

Il n'en demeure pas moins que le même prétraitement devra ultérieurement, sur la base des résultats obtenus dans ce travail, pouvoir être appliqué pour chaque couche de représentation afin de faire des données en entrée de nouveaux échantillons dans un espace de haute dimensionnalité dans lequel il devient possible de travailler sous l'hypothèse

de normalité et ainsi construire des primitives gaussiennes. C'est aussi dans cette phase de prétraitement que l'information sur la résolution temporelle est intégrée au nouvel échantillon.

4.1.1 Synchronisme des sources de données

Toujours dans l'optique de s'affranchir de la technique d'observation, un premier prétraitement simple est effectué relativement à la fréquence d'échantillonnage des données bidimensionnelles.

Puisque les flux de données bidimensionnelles relativement aux agents mobiles peuvent être synchrones ou non, il convient de forcer le traitement sur une base synchrone¹. Advenant qu'il y ait des données manquantes dans le flux de données bidimensionnelles, des données factices sont introduites. Afin de distinguer ces données factices, un attribut binaire de statut est utilisé. Cet attribut est pris en considération dans le prétraitement présenté à la section 4.2.1.

4.1.2 Format des données

Le format de données en entrée est $(t, Id, x, y, c1, c2, \dots, ck)$ où t est le temps, Id est l'identificateur de l'agent mobile, (x, y) sont les coordonnées bidimensionnelles de l'agent mobile et $(c1, c2, \dots, ck)$ représentent k attributs supplémentaires (facultatifs) pouvant être passés directement au module de prétraitement afin d'être juxtaposés à la définition du nouvel échantillon. Puisque la plupart des systèmes en entrée ont leur propre format de données, il est généralement nécessaire d'introduire une interface dédiée au format en entrée. Dans le cadre de ce travail, l'interface représentée à la

¹ Une façon de faire serait à partir d'un tampon qui est vidé à une fréquence d'échantillonnage f_s supérieure ou équivalente à la fréquence de la source de données la plus rapide.

Figure 4.2 a été introduite afin de recueillir les données provenant des fichiers de simulation *RoboCup*.

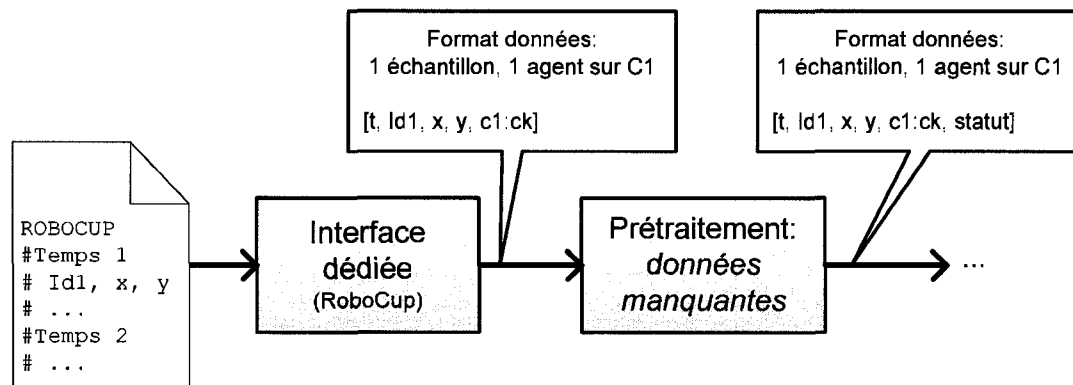


Figure 4.2 : L'interface dédiée convertit le format de données des fichiers RoboCup dans le format (t, Id, x, y, c1, c2, ..., ck) requis.

4.2 L'espace des primitives

L'espace dont il est question ici consiste en un espace de haute dimensionnalité dans lequel il devient possible de décrire une trajectoire en terme de primitives gaussiennes. Il s'agit de l'espace des primitives. La haute dimensionnalité de cet espace a pour objectif de favoriser la séparabilité des observations afin de partitionner cet espace et ainsi réduire la dissimilitude entre l'infinité de trajectoires bidimensionnelles possibles.

4.2.1 Le prétraitement : augmenter la dimensionnalité

La principale opération de prétraitement à effectuer sur les données bidimensionnelles consiste à augmenter la dimensionnalité. La résolution temporelle des données doit notamment être intégrée à ce processus.

D'emblée, il est possible de penser à certaines caractéristiques intuitives telles que la longueur d'une primitive, sa durée ou sa direction résultante susceptibles d'être pertinentes afin de caractériser ladite primitive. Plus généralement, l'idée consiste à construire des histogrammes sur les

attributs relevant de la dynamique en considérant une fenêtre temporelle de grandeur fixée. Les histogrammes, en cumulant quelques échantillons de suite, permettent alors de représenter les attributs tels de petites distributions représentatives de la primitive dans laquelle les échantillons se trouvent.

Pour la couche continue C1, quatre histogrammes sont calculés : le premier sur la direction, les deux suivants sur les déplacements en x et en y , et le quatrième sur la norme de la distance parcourue au carré. La résolution temporelle, pour être considérée, devrait nécessiter l'existence d'un histogramme sur les vitesses, or il est pertinent de remarquer ici que le traitement étant synchrone, l'intervalle de temps entre deux échantillons reste constant ce qui a pour effet de rendre l'histogramme sur les distances équivalent à un autre sur les vitesses. C'est plutôt la présence ou non de données factices qui modifie l'apparence des histogrammes (et du statut bien sûr). En d'autres termes, c'est la représentation dimensionnelle utilisée en terme d'histogrammes qui fournit la base nécessaire à l'intégration de la notion de résolution temporelle. Le schéma *Simulink* des opérations de prétraitement est fourni à l'annexe 3.

Le Tableau 4.1 résume la nouvelle base dimensionnelle définissant l'espace des primitives pour la couche C1. Les histogrammes peuvent suffire à définir la nouvelle base dimensionnelle ou encore être juxtaposés aux attributs déjà existants. L'attribut binaire indiquant le statut de l'échantillon est toujours ajouté à la fin. Afin de pouvoir comparer, les Tableau 4.2 et Tableau 4.3 résumant l'équivalent pour les couches C2 et C3 sont aussi présentés. Dans le cas de C3, il est pertinent de noter que les histogrammes sur les intervalles de temps écoulés sont explicitement ajoutés puisqu'il n'existe plus d'autre information sur la dynamique.

Tableau 4.1 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche continue C1.

Base dimensionnelle enrichie :	#dim :
$(x, y, c_1, \dots, c_k, \theta_1, \dots, \theta_\alpha, \Delta x_1, \dots, \Delta x_i, \Delta y_1, \dots, \Delta y_i, \ d\ _1^2, \dots, \ d\ _i^2, s)$	$k + 2 + \alpha + 3i + 1$
x : Position $\in \mathbb{R}$ selon la première dimension y : Position $\in \mathbb{R}$ selon la seconde dimension c_1, \dots, c_k : k attributs supplémentaires dans le format initial $\theta_1, \dots, \theta_\alpha$: Histogramme des directions ($\alpha = 9$) $\Delta x_1, \dots, \Delta x_i$: Histogramme des déplacements selon la première dimension $\Delta y_1, \dots, \Delta y_i$: Histogramme des déplacements selon la seconde dimension $\ d\ _1^2, \dots, \ d\ _i^2$: Histogramme des distances au carré s : Statut binaire de l'échantillon	

Tableau 4.2 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche discrète C2.

Base dimensionnelle enrichie :	#dim :
$(xId, yId, c_1, \dots, c_k, \theta_1, \dots, \theta_\alpha, \Delta x_1, \dots, \Delta x_i, \Delta y_1, \dots, \Delta y_i, \ d\ _1^2, \dots, \ d\ _i^2, s)$	$k + 2 + \alpha + 3i + 1$
xId : Position selon la première dimension yId : Position selon la seconde dimension c_1, \dots, c_k : k attributs supplémentaires dans le format initial $\theta_1, \dots, \theta_\alpha$: Histogramme des directions ($\alpha = 4$ si maille carrée) $\Delta x_1, \dots, \Delta x_i$: Histogramme des déplacements selon la première dimension $\Delta y_1, \dots, \Delta y_i$: Histogramme des déplacements selon la seconde dimension s : Statut binaire de l'échantillon	

Tableau 4.3 : Résumé de la base dimensionnelle définissant l'espace des primitives pour la couche logique C3.

Base dimensionnelle enrichie :	#dim :
$(zId, c_1, \dots, c_k, zId_1, \dots, zId_i, \Delta t_{11}, \dots, \Delta t_{i1}, \dots, \Delta t_{1j}, \dots, \Delta t_{ij}, s)$	$k + 1 + (j + 1)i + 1$
zId : Identité de la zone logique $\in \mathbb{N}$ courante zId_1, \dots, zId_i : Histogramme des zones parcourues $[\Delta t_1, \dots, \Delta t_i]_j$: Histogrammes des temps écoulés dans chacune des j zones s : Statut binaire de l'échantillon	

4.3 Le modèle graphique probabiliste (PGM)

Les PGMs d'intérêt dans cette recherche sont le HMM et plus particulièrement sa version hiérarchique, le HHMM. L'objectif de cette section est de résumer les concepts principaux et les grandes idées derrière ces deux modèles. D'autres explications sur le lien étroit existant entre la topologie d'un BN et le calcul d'inférence sont fournis à l'annexe 1.

4.3.1 Le modèle markovien caché (HMM)

Les HMMs sont largement répandus et la littérature à ce sujet est fort complète (Bengio, 1999; Ghahramani, 2001; Murphy, 2002). Il est question ici de rappeler à quel genre de questions un HMM peut répondre. Avant tout, il est pertinent de rappeler que les HMMs font partie de la famille des modèles d'espace d'états (*state space model*, SSM) de la même façon que les filtres de Kalman (KFM) et les systèmes linéaires dynamiques (LDS), qui sont tous des exemples de processus stochastiques à temps discret. Il convient généralement de faire l'hypothèse d'invariance temporelle des paramètres, c'est-à-dire que les paramètres du modèle restent les mêmes pour tout le processus. Les KFM et les LDS impliquent des états continus et

se limitent généralement aux cas d'applications linéaires gaussiens. À l'opposé, les HMMs impliquent des états discrets et peuvent être perçus comme l'extension dynamique des modèles de mélange. Puisqu'un modèle de mélange (de gaussiennes) a le potentiel de modéliser une distribution quelconque moyennant un nombre de degrés de liberté suffisant, les HMMs sont largement applicables. Un HMM est souvent interprété comme un automate stochastique dont la dynamique est non linéaire moyennant une matrice de transition appropriée. En supposant qu'il y ait D entités (e.g. D niveaux d'abstraction d'un but) pouvant prendre K valeurs, un HMM aura alors K^D états possibles et il y aura K^{2D} paramètres uniquement pour la matrice de transition.

Plus spécifiquement, une séquence d'observations discrètes ou continues ($\in \mathbb{R}$ afin de simplifier la notation) $y_{1:T} = \{y_1, y_2, \dots, y_t, \dots, y_T\}$ est produite à partir d'une séquence markovienne cachée d'états discrets $Q_{1:T}$. La jointe est

$$P(Y_{1:T}, Q_{1:T}) = P(Y_1 | Q_1) P(Q_1) \prod_{t=2}^T P(Y_t | Q_t) P(Q_t | Q_{t-1})$$

et peut se calculer avec l'algorithme de Baum-Welch (*forward-backward algorithm*) qui est un cas particulier de l'algorithme de propagation de la croyance mentionné à l'annexe 1.

Généralement, pour une séquence d'observations $y_{1:t}$ où l'observation courante se situe à l'instant t , on cherche à évaluer $P(Q_t | y_{1:t})$ (i.e. l'état de la croyance dans le DBN ou *belief state*) avec $\tau = t$, $\tau < t$ ou $\tau > t$. Il en résulte diverses tâches d'inférence dont les distinctions sont énoncées ci-dessous.

A. Filtrage (*filtering*)

Dans ce cas, $\tau = t$ et on trouve

$$\begin{aligned}
P(Q_t | y_{1:t}) &= \frac{1}{P(y_{1:t})} P(y_{1:t}, Q_t) \\
&\propto P(y_t | y_{1:t-1}, Q_t) P(Q_t | y_{1:t-1}) \\
&= P(y_t | Q_t) \sum_{Q_{t-1}} P(Q_t | Q_{t-1}) P(Q_{t-1} | y_{1:t-1})
\end{aligned}$$

soit une étape dite « de prédiction » $P(Q_t | y_{1:t-1})$ et une autre dite « de mise à jour » $P(Q_t | y_{1:t})$. En répétant ces opérations à chaque nouvelle observation, il en résulte un traitement de filtrage en-ligne.

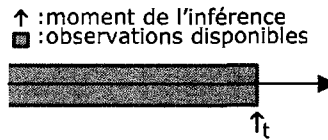


Figure 4.3 : Représentation schématique de la tâche de filtrage.

B. Prédiction (*prediction*)

Dans ce cas, $\tau < t$, c'est-à-dire que $\tau = t - h$ où h représente un pas d'une certaine grandeur dans le futur. En marginalisant sur la prédiction qui est faite de l'état caché Q_{t+h} , il devient possible de prédire de nouvelles observations de la façon suivante

$$P(Y_{t+h} = y | y_{1:t}) = \sum_{Q_{t+h}} P(Y_{t+h} = y | Q_{t+h}) P(Q_{t+h} | y_{1:t})$$

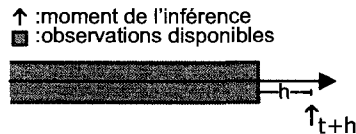


Figure 4.4 : Représentation schématique de la tâche de prédiction.

C. Lissage (*smoothing*)

Dans ce cas, $\tau > t$ et il est plutôt question d'estimer un état passé Q_t considérant toutes les évidences existantes. Il existe deux formes de lissage : le lissage à retard fixe (*fixed-lag smoothing*) et le lissage à intervalle fixe (*fixed-interval smoothing* ou *offline smoothing*).

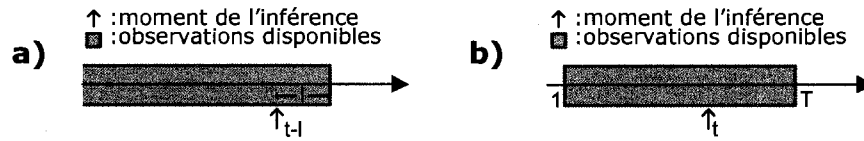


Figure 4.5 : Représentation schématique du lissage à retard fixe (a) et du lissage à intervalle fixe (b).

Dans le premier cas, $\tau = t + l$ où l représente un pas d'une certaine grandeur dans le passé. Ainsi, la tâche d'inférence se résume à calculer $P(Q_{t-l} | y_{1:t})$.

Dans le second cas, le calcul d'inférence est appliqué systématiquement sur des séquences d'observations de longueur fixée. En considérant une séquence comme étant l'ensemble des observations existantes, la tâche se résume à calculer $P(Q_t | y_{1:T})$ qui est essentiellement utilisé pour effectuer l'apprentissage du modèle.

D. Explication la plus probable (*Viterbi decoding*)

Il s'agit ici d'une application de la programmation dynamique permettant de déduire quelle réalisation $q_{1:T}^*$ de la séquence cachée $Q_{1:T}$ est la plus susceptible d'expliquer la séquence $y_{1:T}$ d'observations.

$$\begin{aligned} q_{1:T}^* &= \arg \max_{q_{1:T}} P(Q_{1:T} = q_{1:T} | y_{1:T}) \\ &= \arg \max_{q_{1:T}} P(Q_{1:T} = q_{1:T}, y_{1:T}) \end{aligned}$$

Une représentation schématique de cette tâche est illustrée à la Figure 4.6 où la séquence la plus vraisemblable d'états cachés ressort parmi la multiplicité des séquences possibles.

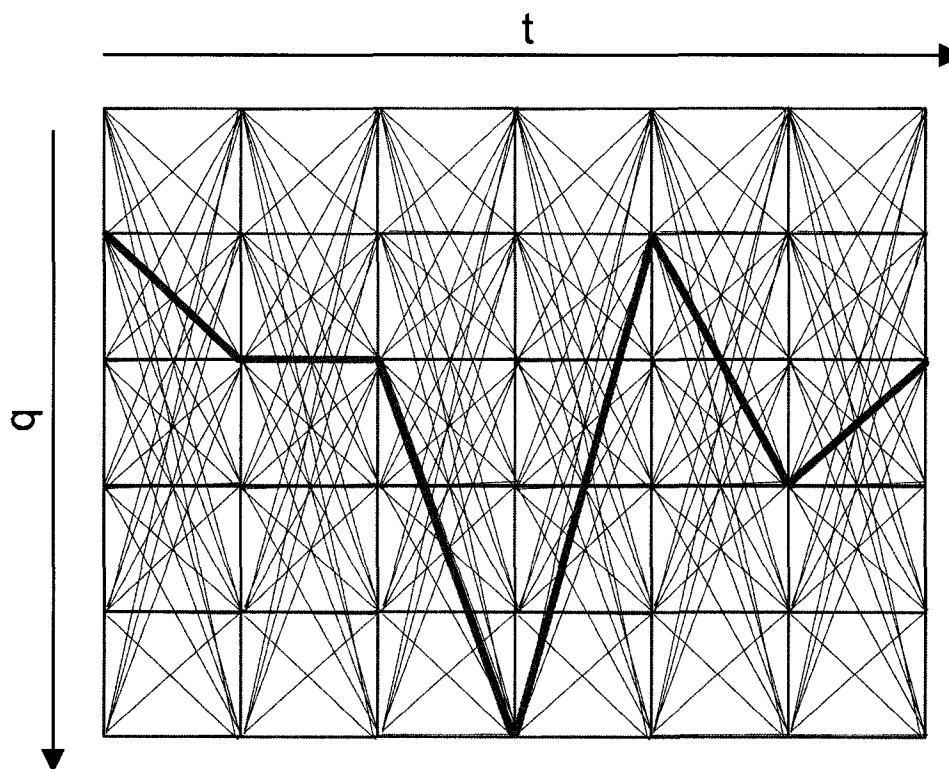


Figure 4.6 : Représentation sous forme de treillis de toutes les séquences possibles dans l'espace des états cachés et, en vert, de la séquence la plus vraisemblable étant donné une séquence d'observations.

4.3.2 Le modèle markovien caché hiérarchique (HHMM)

Tel que mentionné par Murphy (2001), un HHMM est un automate stochastique qui se distingue du HMM par le fait que les états peuvent produire des séquences d'observations en plus des observations singulières. Les états produisant des observations singulières sont dits « état de production » alors que ceux produisant des séquences sont dits « état abstrait ». Chaque état abstrait représente un sous-HHMM pouvant appeler

récurivement d'autres sous-HHMMs créant ainsi une pile d'appels imbriqués. Lorsque qu'un sous-HHMM parvient à sa fin, le contrôle est retourné à l'état abstrait ayant appelé le sous-HHMM en question. Ainsi, tant qu'un sous-HHMM est en cours d'évaluation, l'ensemble des états abstraits à l'origine de ce sous-HHMM (*i.e.* le contexte d'appel) est mémorisé.

Un HHMM peut se représenter sous la forme d'un DBN tel que représenté à la Figure 4.7. En conservant une notation analogue à celle introduite à la section 4.3.1 pour le HMM, l'état de la croyance dans le HHMM s'écrit $P(Q_t^{1:D} | y_{1:t}) = P(Q_t^1, \dots, Q_t^D | y_{1:t})$ où $Q_t^{1:D} = \{Q_t^1, \dots, Q_t^d, \dots, Q_t^D\}$ représente l'empilement du contexte, Q_t^d représente la variable cachée du niveau d'abstraction d à l'instant t , F_t^d représente un indicateur de terminaison sur la séquence $Q_{1:t}^d$ et $y_{1:t}$ représente la séquence d'observations. Il est important de remarquer que le premier niveau de la hiérarchie ($d=1$), c'est-à-dire la racine de la hiérarchie, se situe au haut du DBN alors que le dernier niveau ($d=D$), c'est-à-dire le niveau des états de production, se situe au bas du DBN.

Un HHMM peut être converti en un HMM simplement en créant un état pour chaque configuration possible de l'empilement du contexte $Q_t^{1:D}$. Le HMM aura alors K^D états possibles en supposant que chaque variable Q_t^d puisse prendre K valeurs.

Le HHMM présente toutefois l'avantage de factoriser l'espace des états en D espaces à K partitions, les partitions d'un niveau « appelant » celles des niveaux inférieurs. C'est précisément ce que les arêtes orientées vers le bas entre les variables Q indiquent à la Figure 4.7. Par exemple, le premier niveau divise l'espace en K partitions parmi lesquelles une autre est subdivisée en K plus petites partitions au second niveau, l'une desquelles est à nouveau subdivisée K fois au dernier niveau. Un HHMM présente donc

les avantages d'un arbre de décision dont les décisions à un temps t dépendent de celles de l'arbre à l'instant précédent. Dans le cadre de ce travail, la topologie du DBN est simplifiée à celle représentée à la Figure 4.8 afin de réduire le nombre de paramètres de sorte que les états d'un niveau ne dépendent que des états des niveaux adjacents.

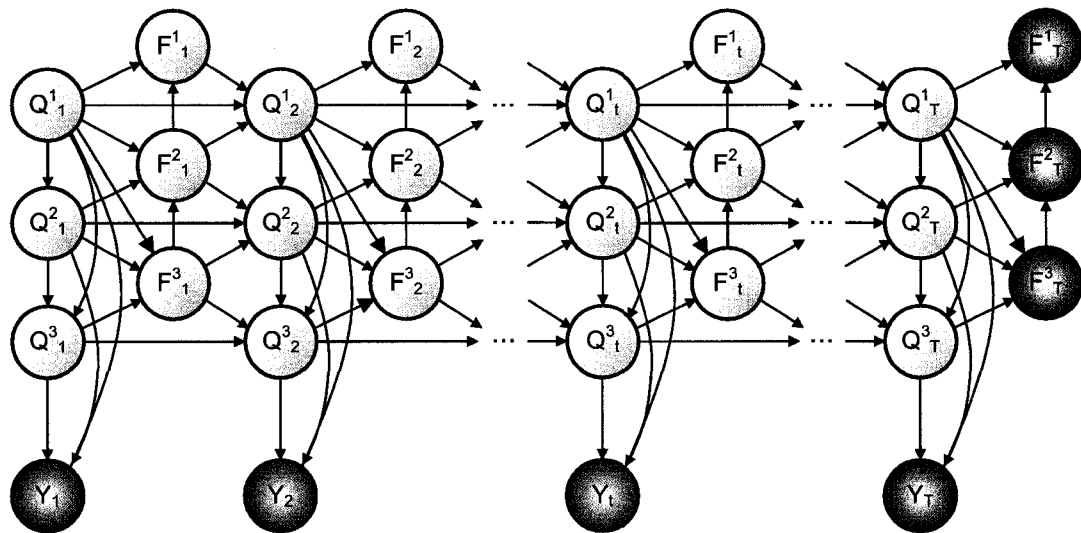


Figure 4.7 : Représentation DBN d'un HHMM à 3 niveaux.

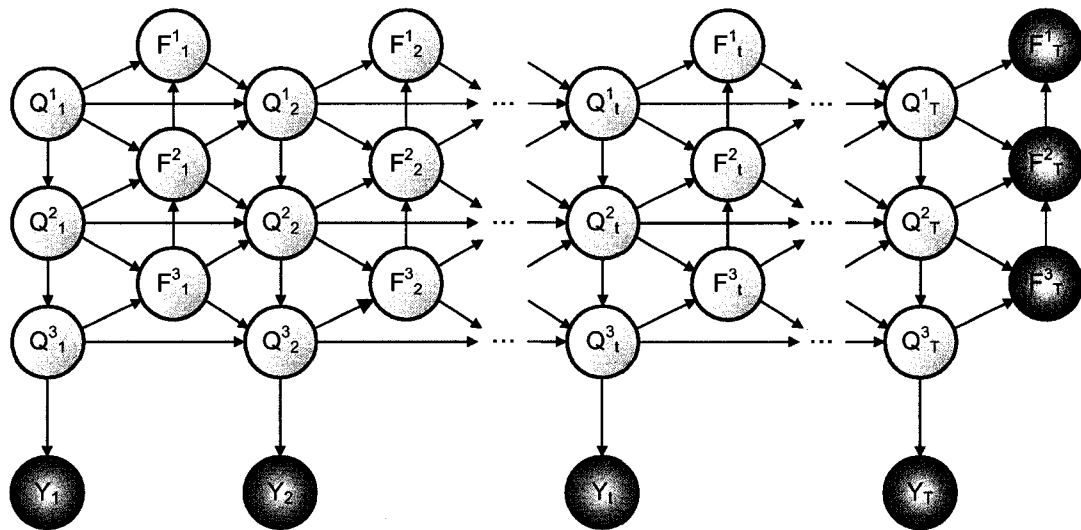


Figure 4.8 : HHMM à 3 niveaux dépendant uniquement du niveau parent.

Il est aussi possible de représenter la topologie d'un DBN (*i.e.* le type et le nombre d'états et les relations d'interdépendance) à partir de son diagramme de transition tel qu'illustré à la Figure 4.9. Dans le cas du HHMM, le diagramme de transition correspond à un arbre dont les sous-structures peuvent (potentiellement) être réutilisées, c'est-à-dire que plusieurs états abstraits peuvent appeler un même sous-HHMM. Cependant, si aucune contrainte n'est ajoutée à la structure, rien n'empêche des états à divers niveaux de changer à un même instant, ce qui n'est clairement pas une éventualité envisageable si le contexte d'appel d'un sous-HHMM doit être mémorisé jusqu'à la fin de ce dernier. C'est précisément pour cette raison que les variables F existent. F_t^d est une variable binaire indiquant que le HMM au niveau d passe dans son état final à l'instant t ($F_t^d = 1$) ou que le HMM est toujours en cours d'évaluation ($F_t^d = 0$). Les arêtes orientées vers le haut entre les variables F forcent les niveaux du haut à évoluer plus lentement que les niveaux du bas. En d'autres termes, un état Q_t^d ne peut changer tant que la séquence d'états Q_t^{d+1} du niveau inférieur n'est pas terminée. Il est intéressant de remarquer que si $F_t^d = 1$, alors $F_t^{d'} = 1$ pour $d' > d$, c'est-à-dire que le HMM du niveau d est en cours d'évaluation et que tous les HMMs des niveaux inférieurs sont terminés. Ainsi, le nombre de variables $F=0$ permet de représenter la hauteur effective du contexte d'appel courant.

En somme, le DBN de la Figure 4.8 est utilisé dans ce travail (avec deux ou trois niveaux) pour reconnaître des primitives (niveau 3) et des patrons de primitives (niveaux 1 et/ou 2) à partir d'observations en haute dimension traitées selon la technique décrite à la section 4.2.1. Ainsi, pour chaque nouvelle observation Y_t , l'identité Q_t^3 de la primitive gaussienne multi-variée qui en est à l'origine est inférée. C'est donc le dernier niveau de la

hiérarchie (niveau 3 dans ce cas-ci) qui détecte les changements de primitives dans le temps, or ces changements sont directement liés à la qualité des observations. Afin de minimiser les transitions superflues possiblement dues à des observations atypiques, une amélioration peut être apportée en ajoutant un modèle de segmentation en-ligne à l'entrée du DBN. Ce modèle de segmentation est discuté à la prochaine section et a pour but de détecter directement les primitives de sorte que les observations ne correspondent plus aux échantillons d'une primitive mais plutôt à l'identité de cette dernière.

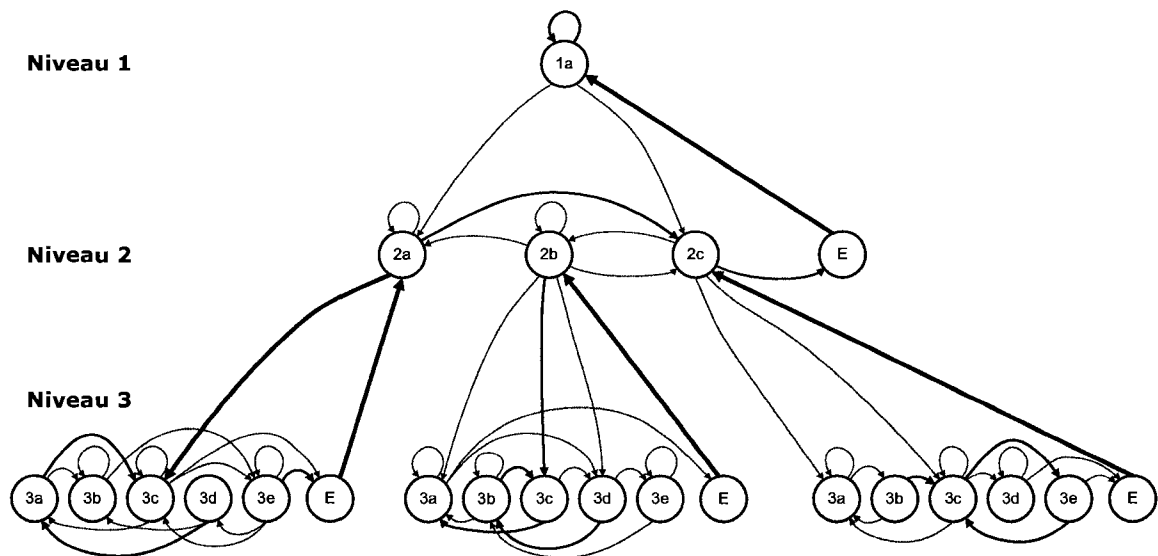


Figure 4.9 : Diagramme de transition représentant la topologie d'un HHMM à trois niveaux. Les nœuds représentent les états alors que les flèches représentent les transitions possibles. L'épaisseur d'une flèche est proportionnelle au poids de sa probabilité. Le premier niveau possède un seul état, le second, quatre alors que le dernier niveau en a cinq. Il est pertinent de remarquer l'existence de l'état « E » indiquant la fin d'une séquence d'états et le retour à son état parent. Cette forme de contrainte est due à l'existence des variables F dans le DBN.

4.4 Le modèle de segmentation en-ligne

Le HHMM est théoriquement en mesure d'inférer les activités d'un agent mobile à partir des primitives reconnues via un modèle d'observation gaussien. Alors pourquoi considérer l'ajout d'un modèle de segmentation?

En premier lieu, un modèle de segmentation permet de réduire la charge de calcul sur le HHMM puisqu'en spécifiant les limites des primitives, le nombre d'inférences à effectuer sur le HHMM diminue. En effet, une seule inférence au moment correspondant à la jonction entre deux primitives peut produire les mêmes résultats qu'une inférence à chaque nouvel échantillon. Si le modèle de segmentation est efficace (en terme de coût de calcul), le processus de segmentation devrait être moins coûteux qu'une segmentation et/ou classification à partir du DBN.

En second lieu, un modèle de segmentation pourrait permettre de caractériser les données en entrée et de les modéliser sous forme de distribution probabiliste, fournissant du coup un judicieux *a priori* pour limiter le sur-apprentissage au niveau du modèle d'observation.

En troisième lieu, un modèle de segmentation pourrait servir à réduire la dimensionnalité des données et ainsi filtrer le bruit avant le modèle d'observation.

Le modèle de segmentation en-ligne repose sur le paradigme de la réduction de la dimensionnalité à partir de techniques simples et efficaces (en terme de coût de calcul) telles que l'analyse en composantes principales (PCA) et sa version probabiliste (PPCA). Il s'agit de deux techniques aptes à détecter des variétés affines dans un espace de haute dimensionnalité tel que l'espace des primitives (*cf.* détails à l'annexe 2). Ainsi, l'idée est d'utiliser la similarité des variétés affines comme critère de base pour la détection de primitives. Les primitives ne sont donc pas de longueur ou de durée fixe, mais plutôt regroupées en fonction de leur uniformité/similarité.

Dans le cas de PCA, deux critères d'uniformité sont considérés :

Critère d'uniformité (dimensionnalité) : « une transition d'une primitive à une autre coïncide avec la donnée où, pour une dimensionnalité M fixée, l'erreur augmente significativement. »

Critère d'uniformité (base vectorielle) : « une transition d'une primitive à une autre coïncide avec la donnée où, pour une base vectorielle fixée, l'erreur augmente significativement. »

Le nombre de compartiments dans les histogrammes introduits dans le prétraitement affecte directement la dimensionnalité de l'espace des primitives et, par conséquent, s'avère primordial dans le processus de segmentation en-ligne. Un nombre de compartiments trop petit limite le discernement des primitives alors qu'un nombre trop grand est sujet à la malédiction de la dimensionnalité et ouvre la porte au sur-apprentissage.

Les travaux considérés dans cette section sont basés sur les modèles de segmentation présentés dans Barbic et al. (2004). Le modèle PPCA est présenté dans Tipping & Bishop (1999) alors que des développements similaires sont introduits par Roweis (1998) sous le nom de PCA « sensible ».

4.4.1 Technique de segmentation PCA (critère dimensionnalité)

Ce modèle de segmentation en-ligne utilise PCA et le critère d'uniformité sur la dimensionnalité. L'idée derrière le modèle réside dans le fait qu'une primitive uniforme peut s'exprimer dans une base de dimensionnalité fixée tant et aussi longtemps que l'erreur générée par l'arrivée de nouvelles données augmente linéairement lorsqu' évaluée avec PCA dans cette même dimensionnalité. Un changement de complexité dans la primitive devrait être reflété par un changement de dimensionnalité, donc par une variation significative de l'erreur.

Le fonctionnement de l'algorithme est sommairement décrit dans les paragraphes qui suivent. Un diagramme fonctionnel complet (en anglais) est fourni à l'annexe 2.

Une primitive est formée d'un ensemble d'échantillons successifs dans le temps. Un nouvel échantillon est ajouté à la primitive tant que l'erreur calculée augmente linéairement.

Une primitive doit contenir un minimum de k échantillons avant que PCA puisse être appliquée et un minimum de l (avec $l < k$) échantillons supplémentaires pour calculer une statistique permettant de détecter une coupure. Par conséquent, les primitives doivent toutes avoir une longueur minimale de $k + l$ échantillons.

À la k ième itération, les valeurs propres $\lambda_j, j = 1, \dots, M$ sont utilisées pour trouver la dimensionnalité M' la plus petite pour laquelle le ratio (4.1) est inférieur à une valeur τ ($0 \leq \tau \leq 1$). À chaque itération $i > k$, PCA est réévaluée et la somme des erreurs au carré sse_i est calculée en utilisant l'expression (9.6).

Selon l'hypothèse principale derrière ce modèle, la dérivée discrète $dsse_i = sse_i - sse_{i-1}$ devrait être quasi-constante (avec de petites oscillations autour de la moyenne) pour une primitive uniforme. Une coupure est introduite lorsque $dsse_i$ est éloignée de $k_\sigma \simeq 3$ écarts-types par rapport aux itérations précédentes.

$$R_{M'} = \frac{\sum_{j=1}^{M'} \lambda_j}{\sum_{j=1}^M \lambda_j} \quad (4.1)$$

4.4.2 Technique de segmentation PCA (critère base vectorielle)

Ce modèle de segmentation en-ligne utilise PCA et le critère d'uniformité de la base vectorielle. L'idée derrière le modèle réside dans le fait qu'une primitive uniforme peut être représentée par une variété dans une base de haute dimensionnalité. Ainsi, une base fixée est appropriée pour représenter une primitive tant et aussi longtemps que l'erreur générée par l'arrivée de nouvelles données augmente linéairement lorsqu'évaluée avec PCA dans cette même base. Un changement de complexité dans la primitive devrait être reflété par une rotation de la base voire même un changement de dimensionnalité, donc par une variation significative de l'erreur.

Le fonctionnement de l'algorithme est similaire à celui décrit à la section précédente à la différence qu'à la k ème itération, la base vectorielle est fixée avec PCA en plus du calcul des valeurs propres. À chaque itération $i > k$, les échantillons sont projetés dans la nouvelle base vectorielle et les $M - (M'+1)$ dernières coordonnées réduites de chaque donnée sont utilisées pour calculer l'erreur de reconstruction sse_i qui, dans ce cas-ci, est calculée à partir de l'expression suivante

$$\begin{aligned} sse &= \frac{1}{2} \sum_{n=1..N} \|\mathbf{x}^n - \hat{\mathbf{x}}^n\|^2 \\ &= \frac{1}{2} \sum_{n=1}^N \sum_{i=M'+1}^M (a_i^n - b_i)^2 \end{aligned} \tag{4.2}$$

où $\mathbf{x}^n, n=1, \dots, N$ est le n ième échantillon $\mathbf{x} \in \mathbb{R}^D$ parmi un ensemble en contenant un total de N et a_i^n et b_i sont respectivement définis par les expressions (9.4) et (9.3). Un diagramme fonctionnel complet (en anglais) est fourni à l'annexe 2.

4.4.3 Technique de segmentation PPCA

Ce modèle de segmentation en-ligne utilise la version probabiliste de PCA pour caractériser les échantillons par un modèle probabiliste gaussien. L'idée derrière le modèle (de segmentation) réside dans le fait qu'une primitive uniforme peut être représentée par une distribution gaussienne multi-variée dont les dimensions superflues (les composantes ne figurant pas comme « principales » au sens PCA) sont remplacées par un bruit (variance sphérique). Il s'agit donc d'un critère d'uniformité sur la distribution gaussienne (donc implicitement sur la dimensionnalité et la base vectorielle) permettant d'estimer la vraisemblance d'un nouvel échantillon d'appartenir à ladite distribution. La distribution reste appropriée pour décrire un certain nombre d'échantillons tant et aussi longtemps que la statistique d'appartenance à la distribution, c'est-à-dire la distance de Mahalanobis moyenne (définie plus bas) des échantillons par rapport à la distribution, est inférieure à un certain seuil.

Le fonctionnement de l'algorithme est sommairement décrit dans les paragraphes qui suivent. Un diagramme fonctionnel complet (en anglais) est fourni à l'annexe 2.

Une primitive est formée d'un ensemble d'échantillons successifs dans le temps. Le modèle probabiliste est évalué à chaque fois qu'un certain nombre d'échantillons est atteint tant et aussi longtemps qu'aucune coupure n'est insérée. La statistique d'appartenance à la distribution est la distance de Mahalanobis H telle que

$$H = \frac{1}{m} \sum_{i=k+1}^{k+m} (\mathbf{x}^n - \bar{\mathbf{x}})^T \mathbf{C}^{-1} (\mathbf{x}^n - \bar{\mathbf{x}}) \quad (4.3)$$

où k est le nombre d'échantillons ayant servi à évaluer le modèle probabiliste et $\bar{\mathbf{x}}$ et \mathbf{C} sont respectivement la moyenne et la matrice de covariance du modèle probabiliste gaussien. La matrice de covariance peut être calculée comme suit

$$\mathbf{C} = \frac{1}{k-1} \mathbf{U} \tilde{\Lambda} \mathbf{U}^T \quad (4.4)$$

à partir de la matrice \mathbf{U} dont les colonnes ordonnées selon les valeurs propres décroissantes représentent les vecteurs propres de la nouvelle base et de la matrice diagonale $\tilde{\Lambda}$ dont les M' premières valeurs sur la diagonales représentent les M' principales valeurs propres décroissantes de Σ et les autres $(M - M')$ valeurs sur la diagonale sont remplacées par un bruit sphérique λ équivalent à la moyenne des valeurs propres écartées

$$\lambda = \frac{1}{M - M'} \sum_{i=M'+1}^M \lambda_i \quad (4.5)$$

Il faut un minimum de m échantillons pour calculer la statistique énoncée en (4.3). Préalablement, un minimum de m échantillons est aussi nécessaire pour évaluer le modèle probabiliste, c'est-à-dire pour calculer la moyenne $\bar{\mathbf{x}}$ et la covariance \mathbf{C} . La procédure pour déterminer la dimensionnalité M' optimale fait appel au ratio (4.1) de la même façon que dans les modèles de segmentation basés sur PCA.

Ainsi, les m premiers échantillons servent à la première évaluation du modèle probabiliste et les m échantillons qui suivent permettent de calculer la première statistique.

Le modèle probabiliste est réévalué avec inc échantillons supplémentaires (où inc est environ 5–10% de m) et la statistique est réévaluée avec les m échantillons qui suivent. Le tout est répété tant que la statistique ne dépasse pas un certain seuil.

Pour une primitive de longueur L , la statistique atteint un maximum lorsque les L échantillons ont servi à évaluer le modèle probabiliste et que les m échantillons qui suivent (soient les m premiers de la nouvelle primitive) servent à calculer la statistique. Avec les inc échantillons suivants, le

modèle commence à accommoder la nouvelle primitive et une coupure est insérée. À l'inverse, la statistique tend vers son minimum lorsque le modèle probabiliste est à son plus précis, c'est-à-dire évalué avec les $\sim L - m$ derniers échantillons (les échantillons utilisés pour évaluer la statistique font toujours partie de la même distribution). Il va donc de soi que la longueur minimale d'une primitive soit $L = 2m$.

Selon l'hypothèse d'uniformité de la distribution derrière ce modèle, l'évolution de la statistique dans le temps devrait passer par un minimum local puis par un maximum. Si la différence entre les deux extrêmes dépasse le seuil fixé (en terme de nombre d'écarts-types), alors une coupure est introduite. S'il advient que la statistique passe par plusieurs minima et/ou maxima locaux, alors il est probable que les hyper-paramètres m et inc soient trop grands ou que les primitives ne soient pas gaussiennes.

Tous les aspects nécessaires à l'édification du modèle d'activité ont été abordés. La modélisation multi-résolutions, l'espace des primitives, les opérations de prétraitement, les bases théoriques derrière les BNs, HMM et HHMM et les modèles de segmentation en-ligne ont tous été exposés. La méthodologie d'expérimentation est explicitée au chapitre suivant puis suivent les diverses expérimentations aux chapitres 6 et 7.

CHAPITRE 5 : « LE PLAN DE MATCH »

La méthodologie d'expérimentation est ici présentée selon deux approches distinctes : l'une dans laquelle le PGM est utilisé seul, c'est-à-dire directement sur l'ensemble des échantillons, et l'autre dans laquelle le PGM est utilisé conjointement avec le modèle de segmentation en-ligne présentant les meilleurs résultats lors de tests préalables. Dans les deux approches, trois contextes d'application sont considérés. Ces contextes d'application sont d'abord introduits.

5.1 Les trois contextes d'application

Trois contextes d'application ont été modélisés afin de pouvoir tester le modèle d'activité par paliers, c'est-à-dire dans des contextes dont la complexité augmente à chaque fois. Les trois contextes d'application sont détaillés dans les sections qui suivent.

Des séquences aléatoires de primitives gaussiennes et de primitives de type trajectoire bidimensionnelle sont considérées dans les deux premiers contextes d'application. Finalement, des trajectoires bidimensionnelles provenant de réelles compétitions *RoboCup* sont utilisées dans le troisième contexte d'application.

Dans les deux premiers contextes d'application, les séquences sont générées à partir de prémisses connues afin que les résultats puissent être validés. Le troisième contexte se veut plus descriptif et exploratoire puisqu'il n'existe aucune information disponible quant aux résultats désirés.

5.1.1 Séquences (abstraites) de primitives gaussiennes

Dans ce contexte d'application, un générateur aléatoire de primitives gaussiennes multi-variées permet de générer des séquences afin de caractériser la performance du PGM en situation idéale et de valider aisément le modèle de segmentation en-ligne.

En effet, puisque le modèle d'observation du PGM est gaussien et que le modèle de segmentation, PCA ou PPCA, repose sur des prémisses ou des propriétés qui relèvent aussi des gaussiennes, l'utilisation de primitives générées directement à partir de distributions gaussiennes multi-variées devrait permettre aux modèles d'atteindre leur performance optimale. Par conséquent, les séquences générées sont nécessairement abstraites et ne correspondent pas à des primitives provenant de réelles trajectoires bidimensionnelles.

Le générateur est conçu de sorte qu'il est possible de tout générer aléatoirement ou encore de spécifier certaines prémisses telles que la dimensionnalité, le nombre de primitives, les divers paramètres des gaussiennes (exactement comme un mélange de gaussiennes, *i.e.* poids statistique, centre et covariance), les bornes sur la longueur de chaque primitive, le patron de la séquence, etc. Un exemple, illustré à la Figure 5.1, représente quatre (4) séquences d'environ 200 échantillons générés à partir de patrons quelconques formés d'au plus quatre (4) primitives tridimensionnelles parmi une possibilité de cinq (5).

De plus, le prétraitement en haute dimension est omis, permettant du coup d'éliminer une autre source d'indétermination. En effet, l'objectif du prétraitement est d'enrichir la dimensionnalité afin d'obtenir des variétés gaussiennes (les primitives), or les primitives générées ici sont déjà gaussiennes.

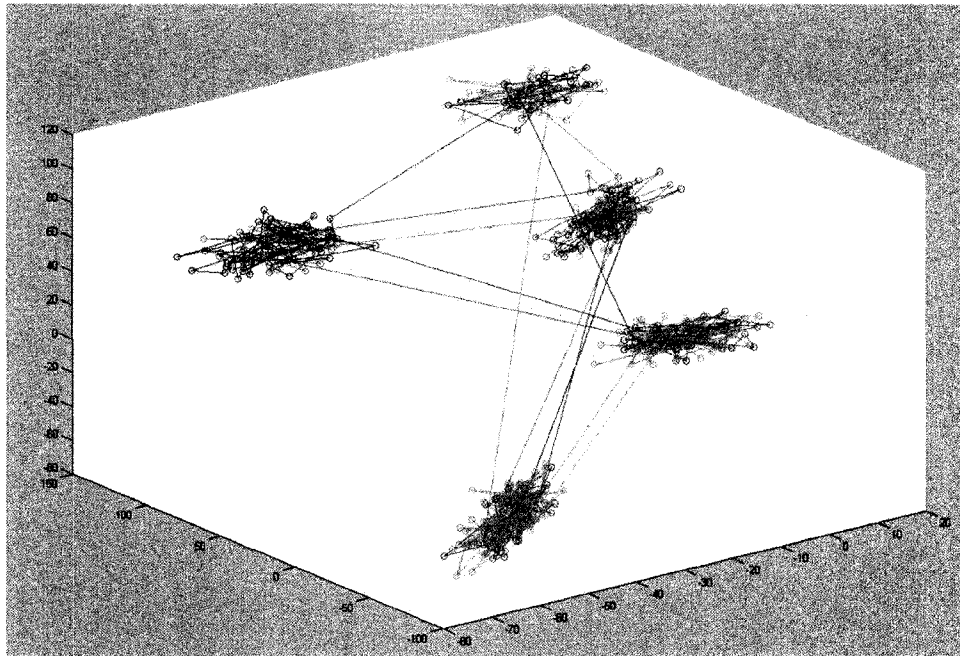


Figure 5.1 : Quatre séquences d'environ 200 échantillons provenant de patrons quelconques impliquant au plus quatre primitives gaussiennes tridimensionnelles parmi une possibilité de cinq.

5.1.2 Séquences (aléatoires) de primitives 2D

Dans ce contexte d'application, un générateur aléatoire de primitives de type trajectoire permet de générer des trajectoires bidimensionnelles dont les prémisses sont connues afin de valider le PGM et le modèle de segmentation en-ligne. Puisqu'il s'agit ici de trajectoires bidimensionnelles, le prétraitement en haute dimension doit être appliqué aux échantillons générés.

Le générateur est conçu de sorte qu'il est possible de générer des primitives de sept types différents : « immobile », « rectiligne à vitesse constante », « rectiligne avec accélération », « rectiligne avec décélération », « curviligne dans le sens horaire », « curviligne dans le sens antihoraire » ou « aléatoire ».

Dans chaque cas, un bruit gaussien est toujours ajouté à la trajectoire. Il est aussi possible de spécifier certaines prémisses telles que les bornes sur la longueur de chaque primitive, le patron de la séquence, la vitesse moyenne, la vitesse maximale, la fréquence d'échantillonnage, etc. Deux exemples représentant sept (7) trajectoires d'environ 200 échantillons générés à partir de deux (2) patrons distincts sont illustrés aux Figure 5.2 et Figure 5.3.

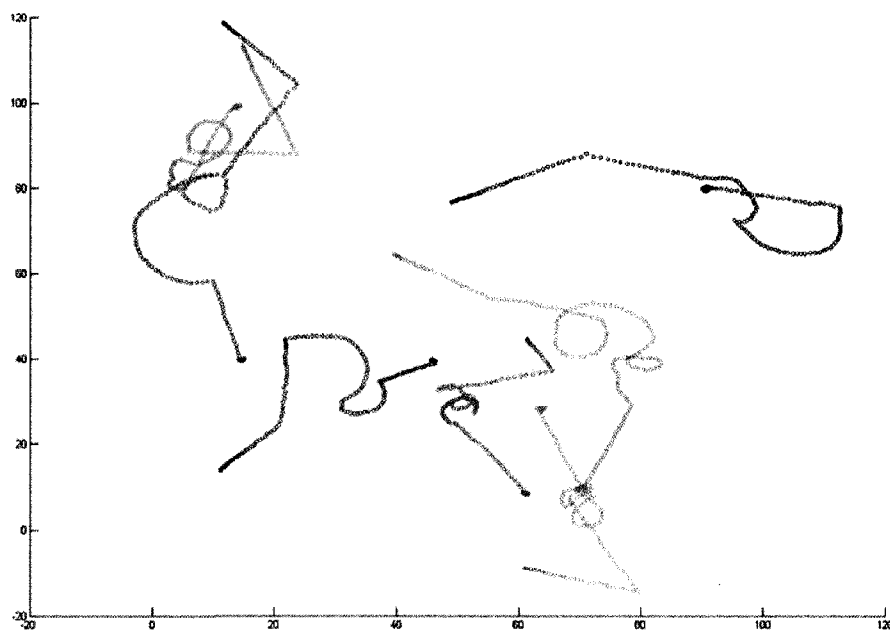


Figure 5.2 : Sept trajectoires bidimensionnelles d'environ 200 échantillons provenant du patron de primitives de type trajectoire « immobile », « rectiligne à vitesse constante », « curviligne dans le sens horaire », « curviligne dans le sens antihoraire », « rectiligne avec accélération » et « rectiligne avec décélération ».

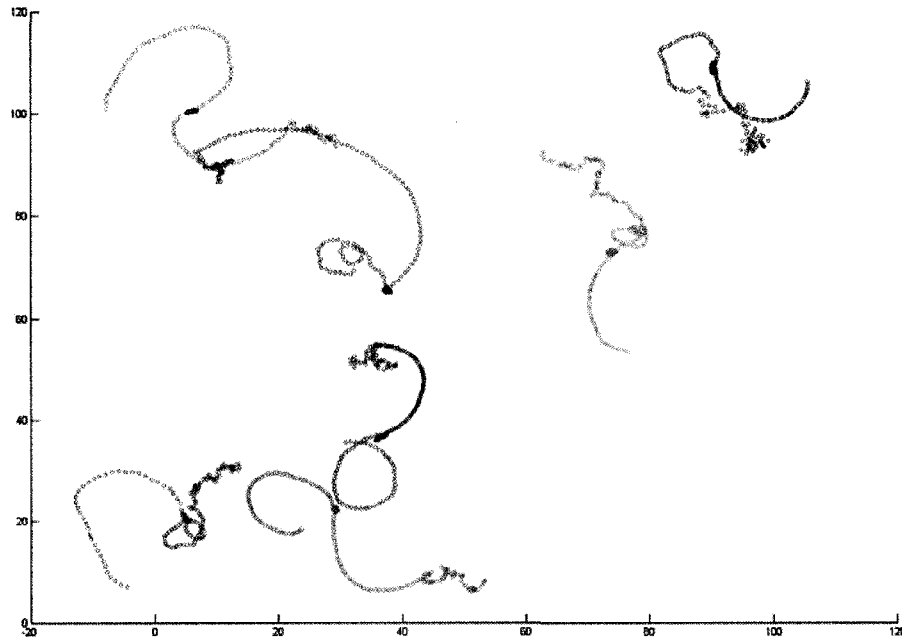


Figure 5.3 : Sept trajectoires bidimensionnelles d'environ 200 échantillons provenant du patron de primitives de type trajectoire « aléatoire », « curviligne dans le sens horaire », « immobile » et « curviligne dans le sens antihoraire ».

5.1.3 Trajectoires réelles (RoboCup)

Dans ce contexte d'application, les trajectoires bidimensionnelles de joueurs de soccer robotisés prenant part à une compétition de la ligue de simulation de *RoboCup* sont directement utilisées. Aucune connaissance *a priori* n'est disponible de sorte que les résultats ont une portée exploratoire et descriptive. Les modèles considérés dans ce contexte d'application doivent nécessairement être préalablement validés dans les deux autres contextes d'application. Puisqu'il s'agit ici de trajectoires bidimensionnelles, le prétraitement en haute dimension doit être appliqué aux échantillons observés.

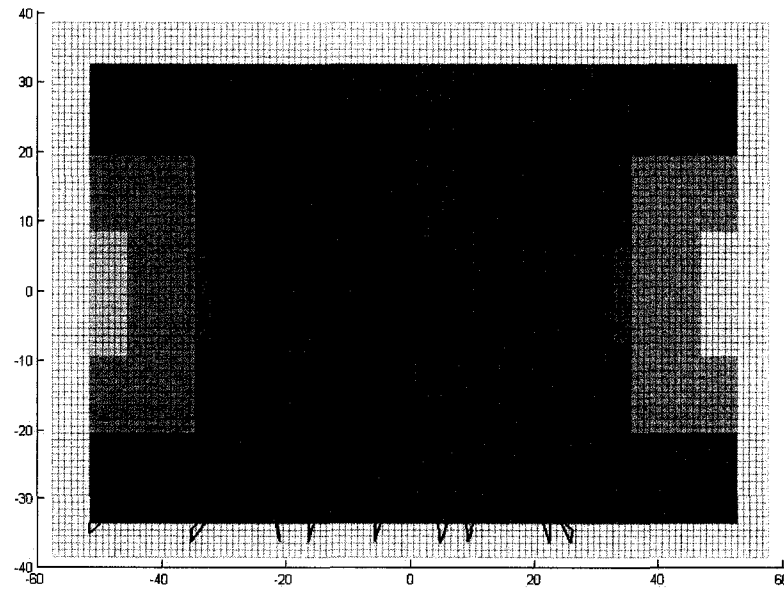


Figure 5.4 : Trajectoire bidimensionnelle d'un joueur de soccer robotisé durant une partie complète d'une compétition RoboCup.

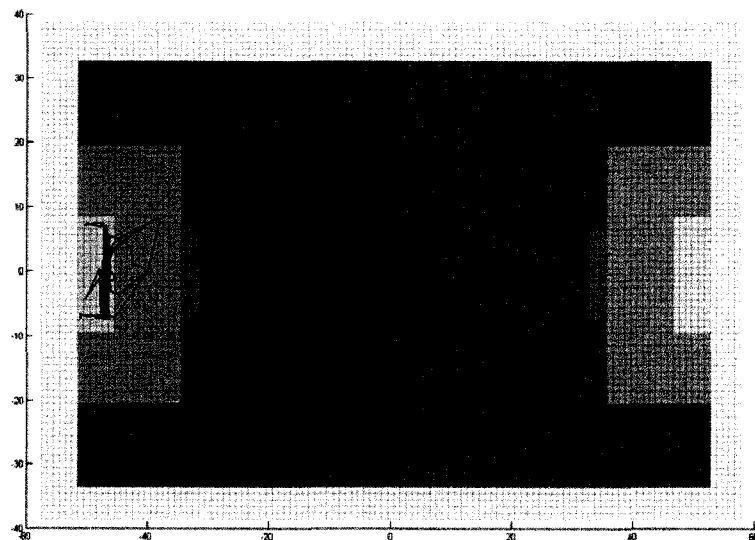


Figure 5.5 : Trajectoire bidimensionnelle d'un gardien de but robotisé durant une partie complète d'une compétition RoboCup.

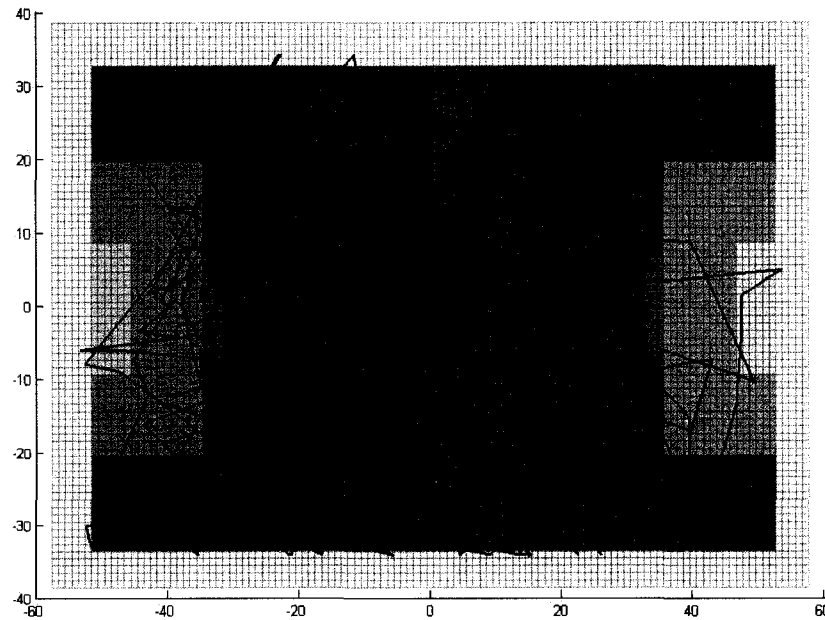


Figure 5.6 : Trajectoire bidimensionnelle du ballon durant une partie complète d'une compétition RoboCup.

Les Figure 5.4, Figure 5.5 et Figure 5.6 représentent respectivement les trajectoires bidimensionnelles d'un joueur, du gardien et du ballon pendant une partie complète.

5.2 Méthodologie de test : PGM seul

Il convient maintenant de décrire la méthodologie de test pour le PGM lorsque l'ensemble des échantillons lui est présenté en-ligne, c'est-à-dire les uns après les autres sans aucune segmentation préalable et ce, pour chacun des trois contextes d'application introduits ci-dessus. Le modèle d'observation du PGM est continu de type gaussien conditionnel. Les matrices de covariance des composantes gaussiennes sont considérées complètes (par opposition aux matrices diagonales ou sphériques).

De façon générale, la méthodologie est divisée en cinq sections : « objectif général », « situation », « stratégie de test », « méthode de test » et « résultats attendus ».

5.2.1 Séquences (abstraites) de primitives gaussiennes

Objectif général Vérifier la capacité d'un HMM et d'un HHMM à trouver l'explication la plus probable de séquences de primitives gaussiennes et décrire les CPT obtenus.

Situation Il est d'abord pertinent de vérifier qu'un HHMM parvient au minimum aux mêmes résultats qu'un HMM. De plus, sachant que des séquences (abstraites) sont générées selon des patrons connus contenant un sous-patron lui aussi connu, un HHMM est-il plus susceptible de reconnaître les sous-séquences qu'un HMM? Puisque les primitives gaussiennes sont connues, le prétraitement en haute dimension est court-circuité.

Stratégie de test Considérer un HMM et un HHMM, la topologie du HHMM étant identique au HMM mais avec un niveau de plus. Comparer les deux modèles sur les mêmes séquences.

Pour chaque séquence de test, trouver l'explication la plus probable (du dernier niveau) avec Viterbi. Coter le PGM en terme du pourcentage de la séquence correctement reconnu. Aussi, pour chaque couche cachée, calculer la marginale $P(Q_t | y_{1:t})$ pour $1 \leq t \leq T$ où $y_{1:t}$ représente la séquence des échantillons dans l'espace des primitives jusqu'à l'instant t .

Méthode de test Pour chaque PGM, considérer 25 séquences de longueur variable (entre 60 et 150 échantillons) générées à partir de primitives gaussiennes. Ces séquences contiennent entre trois (3) et cinq (5) primitives parmi lesquelles il doit y avoir une (1) réalisation d'un sous-patron de trois (3) primitives pour 75-80% des séquences. Fixer le nombre d'états à quatre (4) pour le dernier niveau des PGMs. Fixer le nombre d'états du niveau supérieur à deux (2). Considérer des dimensionnalités de trois (3) et 50. Entraîner avec ces 25 séquences. Puisque l'algorithme EM est sensible aux minima locaux, envisager plusieurs répétitions avec

différentes valeurs d'initialisation. Pour ce faire, initialiser les moyennes des gaussiennes à partir d'un sous-ensemble d'échantillons. Tester avec cinq (5) nouvelles séquences issues du générateur.

Résultats attendus Les deux PGMs devraient être en mesure de reconnaître la presque totalité des primitives. Le HHMM devrait être en mesure reconnaître l'existence d'un sous-patron via le niveau hiérarchique supplémentaire.

5.2.2 Séquences (aléatoires) de primitives 2D

Il s'agit ici de la même méthodologie qu'à la section précédente, à la différence qu'il doit y avoir un prétraitement en haute dimension puisqu'il s'agit de séquences de type trajectoire bidimensionnelle. Des primitives dont la longueur varie entre 80 et 150 échantillons sont considérées.

Résultats attendus Les deux PGMs devraient avoir un rendement moins élevé qu'avec les primitives gaussiennes. L'explication la plus probable devrait être plus bruitée étant donné la présence d'états atypiques (*outliers*) provenant probablement de primitives quasi gaussiennes voire même non gaussiennes. La grande dimensionnalité obligée (d'environ 50) des échantillons pourrait engendrer un sur-apprentissage qui devrait être rectifié par un ajustement du modèle bayésien.

5.3 Méthodologie de test : PGM avec modèle de segmentation en-ligne

Il convient maintenant de décrire la méthodologie de test pour le PGM lorsqu'utilisé avec le modèle de segmentation en-ligne, c'est-à-dire sur des primitives déjà segmentées et ce, pour chacun des trois contextes d'application introduits ci-dessus. La méthodologie est divisée de la même façon qu'à la section précédente.

5.3.1 Séquences (abstraites) de primitives gaussiennes

A. Le modèle de segmentation seul

Objectif général Vérifier la capacité des modèles de segmentation PCA (critère de dimensionnalité et de la base vectorielle) et PPCA à trouver correctement le commencement et la fin des primitives dans des séquences de primitives gaussiennes.

Situation Afin de valider l'aptitude des modèles de segmentation en-ligne à correctement segmenter des séquences de primitives sans avoir besoin de spécifier les coupures manuellement et aussi afin d'éliminer les indéterminations pouvant provenir du prétraitement en haute dimension, il convient d'utiliser un générateur de quelconques séquences (abstraites) générées selon des patrons connus.

Stratégie de test Considérer les trois modèles de segmentation en-ligne et explorer diverses valeurs d'hyper-paramètres pour chaque modèle. Coter les modèles de segmentation par un taux de détection (rapport du nombre de primitives détectées sur le nombre total de primitives à détecter) et un taux de succès (rapport du nombre de primitives correctement détectées sur le nombre total de primitives détectées). Calculer à partir de ces deux taux le taux de primitives détectées. Soient n_{prim} le nombre de primitives distinctes à détecter dans la séquence, $ncuts$ le nombre de détections/coupures, et $nsuccess$ le nombre de détections (ou coupures) valides, c'est-à-dire se situant dans une fenêtre de n échantillons autour de la jonction entre deux primitives, alors le taux (ratio) de détection est

$$R_c = \frac{ncuts}{n_{prim}} \quad (5.1)$$

et le taux (ratio) de succès est

$$R_s = \frac{nsuccess}{ncuts} \quad (5.2)$$

Le taux de primitives détectées est

$$R_c R_s = \frac{nsuccess}{nprim} \quad (5.3)$$

Le score du modèle est alors défini par l'expression (5.4) comme une valeur entre 0 et 1 où l'unité indique le meilleur modèle, c'est-à-dire celui avec le plus de détections valides et limitant le nombre de détections superflues.

$$score = \begin{cases} \frac{2(R_c R_s)}{R_c + R_s}, & \text{si } R_c > 0 \text{ et } R_s > 0 \\ 0 & \text{, si } R_c = 0 \text{ et/ou } R_s = 0 \end{cases} \quad (5.4)$$

Situer les modèles testés par rapport à une segmentation à temps fixe fréquente et une autre peu fréquente.

Méthode de test Pour chaque type de modèle, considérer environ cinq (5) configurations d'hyper-paramètres judicieusement choisis (*i.e.* choisis relativement à une première phase exploratoire). Générer dix (10) séquences de quatre (4) primitives gaussiennes multi-variées (environ 150 échantillons par primitive) dans des espaces à 25 et 70 dimensions. Considérer le cas où les primitives gaussiennes sont clairement éloignées les unes des autres (*i.e.* séparées d'au moins 2,6 écarts-types) et le cas où elles se chevauchent (*i.e.* séparées d'environ 1 écart-type). Considérer le cas où toutes les primitives ont un nombre équivalent de composantes principales et le cas où le nombre de composantes principales est variable.

Résultats attendus De façon générale, tous les modèles testés devraient avoir un rendement supérieur aux résultats obtenus par une segmentation à temps fixe. Les modèles segmentation de type PCA devraient toutefois avoir de la difficulté à distinguer des primitives ayant un nombre

équivalent de composantes principales ou des composantes principales « orientées » selon les mêmes dimensions. Le modèle de segmentation de type PPCA devrait être moins affecté par ce genre de contraintes. Puisqu'il s'agit d'une application du cas idéal gaussien, les divers modèles devraient avoir un rendement optimal, plus particulièrement dans le cas PPCA.

B. Le PGM avec le modèle de segmentation

Il s'agit ici de la même méthodologie qu'à la section 5.2.1 à la différence que les échantillons ne sont pas traités un à un dans le PGM (seul le HHMM est considéré ici), mais plutôt globalement lors d'un changement de primitives. Ainsi, si une séquence contient quatre primitives, alors le PGM traitera uniquement quatre observations (plutôt que plusieurs centaines, voire plusieurs milliers), l'observation étant le résultat d'une classification préalable à partir de tous les échantillons d'une même primitive. Le modèle d'observation du PGM est donc discret (*i.e.* CPT ou CPD multinomiale). Afin de rester dans le même paradigme et par simplicité, un classificateur bayésien naïf est utilisé pour cette tâche. Toutefois, tout autre classificateur capable d'un minimum de calculs à l'exécution, notamment dans le paradigme des approches discriminatives, pourrait être plus approprié pour cette tâche.

5.3.2 Séquences (aléatoires) de primitives 2D

A. Le modèle de segmentation seul

Il s'agit ici de la même méthodologie qu'à la section précédente (5.3.1 A), à la différence qu'il doit y avoir un prétraitement en haute dimension puisqu'il s'agit de séquences de type trajectoire bidimensionnelle.

Résultats attendus De façon générale, les modèles segmentation de type PCA devraient avoir de la difficulté à distinguer des primitives ayant un nombre équivalent de composantes principales ou des composantes principales « orientées » selon les mêmes dimensions comme, par

exemple, une trajectoire circulaire et un attracteur immobile. Le modèle de segmentation de type PPCA devrait être moins affecté par ce genre de contraintes et être plus robuste à un tel contexte d'application. Cependant, puisqu'il s'agit d'un contexte d'application de type trajectoire bidimensionnelle avec prétraitement en haute dimension, le rendement des divers modèles devrait être moindre que dans le cas de primitives gaussiennes, surtout si la normalité des primitives est compromise.

B. Le PGM avec le modèle de segmentation

Il s'agit ici de la même méthodologie qu'à la section 5.2.1 à la différence qu'il doit y avoir un prétraitement en haute dimension puisqu'il s'agit de séquences de type trajectoire bidimensionnelle et que les échantillons ne sont pas traités un à un dans le PGM, mais plutôt globalement lors d'un changement de primitives tel qu'énoncé à la section 5.3.1 B. Les CPTs sont des matrices stochastiques initialisées aléatoirement à partir d'une distribution uniforme.

5.3.3 Trajectoires réelles (RoboCup)

Objectif général Faire une analyse descriptive des résultats/primitives /CPTs obtenus, pour un HHMM, après un apprentissage sur des trajectoires bidimensionnelles de joueurs de soccer robotisés durant une partie d'une compétition *RoboCup*.

Situation Puisqu'il s'agit de trajectoires bidimensionnelles, le prétraitement en haute dimension est nécessaire. Cependant, puisque les primitives d'origine sont inconnues, l'obtention de résultats tels que l'explication la plus probable devient inutile.

Stratégie de test Considérer uniquement un HHMM, toujours avec deux niveaux de hiérarchie.

Méthode de test Pour l'entraînement, considérer une partie complète comme étant équivalente à une foule de trajectoires de longueur $L = 800$

(soit une durée d'environ 1 m 20 s) pouvant se chevaucher. Considérer un joueur à titre d'exemples. Représenter les primitives bidimensionnelles les plus fréquemment reconnues. Fixer le nombre d'états à cinq (5) pour le dernier niveau des PGMs. Fixer le nombre d'états du niveau supérieur à deux (2). Tester avec des trajectoires jamais observées auparavant.

5.3.4 Coût de calcul temps réel

Objectif général Estimer le coût de calcul (CPU/temps) du PGM.

Situation Afin de valider l'aspect « temps réel » du modèle d'activité, il est important d'estimer la charge de calcul au fur et à mesure que la trajectoire bidimensionnelle observée en entrée s'allonge.

Stratégie de test Considérer deux HHMMs, l'un avec deux niveaux de hiérarchie et l'autre avec trois niveaux, appliqués une même longue trajectoire. S'assurer que la dérivée du temps de calcul reste constante et inférieure à 0,1 s (ou 10 Hz).

Méthode de test Pour chaque HHMM, traiter une trajectoire bidimensionnelle de longueur $L = 3000$ (soit une durée d'environ 5 min.) et observer le temps de traitement au fur et à mesure.

Résultats attendus Le temps de traitement pour un échantillon devrait être constant pour les deux HHMMs. Le HHMM à trois niveaux devrait toutefois afficher un temps de traitement significativement plus élevé que le HHMM à deux niveaux, probablement supérieur à la limite acceptable pour le temps réel dans ce contexte d'application, soit 10 échantillons / sec ou 0,1 s par échantillon.

La méthodologie d'expérimentation étant maintenant bien définie dans les trois contextes d'application, il ne reste plus qu'à présenter les résultats des

diverses expérimentations. La première approche, impliquant l'utilisation du PGM seul sur les échantillons, est couverte au chapitre 6 alors que la seconde, dans laquelle le modèle de segmentation en-ligne est ajouté, est couverte au chapitre 7.

CHAPITRE 6 : « LA 1^{ÈRE} DEMIE »

Les résultats des diverses expérimentations impliquant le PGM utilisé directement sur l'ensemble des échantillons sont présentés dans ce chapitre alors que ceux découlant de l'utilisation conjointe du PGM et du modèle de segmentation en-ligne sont présentés au chapitre suivant. La validation du PGM est effectuée dans les trois contextes d'application introduits à la section 5.1. Le coût de calcul en temps réel est aussi présenté.

Il convient de rappeler ici que l'emploi du PGM directement sur les échantillons implique un modèle d'observation gaussien conditionnel permettant d'identifier les primitives à partir du flux d'échantillons. Les CPDs gaussiennes conditionnelles sont paramétrées par un vecteur ($\in \mathbb{R}^M$) représentant la moyenne (*i.e.* le prototype d'une primitive) dans l'espace des primitives, dont la dimensionnalité est M , de même que par une pleine matrice de covariance de dimension $M \times M$.

6.1 Validation du PGM

Afin de représenter les résultats obtenus de façon précise, consistante et objective tout en considérant, dans leur ensemble, les divers PGMs évalués, deux informations sont principalement considérées. En premier lieu, la capacité du PGM à reconnaître une certaine séquence de primitives en entrée est évaluée en comparant l'explication la plus probable (MPE, calculée avec Viterbi) avec l'identité des primitives en entrée. Le pourcentage obtenu en calculant le rapport du nombre d'échantillons correctement reconnus sur la longueur totale de la séquence représente le score du PGM. En second lieu, l'état global du PGM, c'est-à-dire la réalisation de tous les états de l'empilement du contexte $Q_t^{1:D}$, est évalué et représenté graphiquement en marginalisant à chaque instant t les variables Q_t^d y figurant. En d'autres termes, la marginale $P(Q_t^{1:D} | y_{1:t}) = P(Q_t^1, \dots, Q_t^D | y_{1:t})$ pour $1 \leq t \leq T$ est calculée

en-ligne et les probabilités de chaque état sont représentées graphiquement.

6.1.1 Séquences (abstraites) de primitives gaussiennes

Pour parvenir à ces deux informations (scores et marginales), la méthodologie énoncée à la section 5.2.1 est appliquée pour les deux PGMs (HMM et HHMM à deux niveaux). Les scores sont calculés à partir de 25 séquences de test (cinq séquences/patron) dont 20 contiennent une même sous-séquence (4-1-2). Plusieurs répétitions de la phase d'apprentissage sont effectuées avec diverses valeurs d'initialisation pour les paramètres. Les répétitions pour lesquelles la log-vraisemblance est clairement supérieure à celle des autres itérations révèlent l'existence d'un minimum local et sont par conséquent écartées. Les résultats sont compilés dans le Tableau 6.1 pour les primitives en trois dimensions et dans le Tableau 6.2 pour celles en 50 dimensions.

Tableau 6.1 : Scores des deux PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace de primitives gaussiennes à 3 dimensions.

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)
1-(4-1-2)	102	100	100
(4-1-2)-3-4	127	100	99
2-4-3-1	100	99	100
3-(4-1-2)-4	122	100	100
(4-1-2)	71	99	99
	Score moyen	100	100

Tableau 6.2 : Scores des deux PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace de primitives gaussiennes à 50 dimensions.

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)
1-(4-1-2)	96	100	100
(4-1-2)-3-4	122	100	100
2-4-3-1	108	100	100
3-(4-1-2)-4	128	100	100
(4-1-2)	72	100	100
	Score moyen	100	100

Les probabilités marginales dans le temps de chaque état de l'empilement du contexte des deux PGMs sont représentées aux Figure 6.1 et Figure 6.2 associées respectivement à des observations provenant d'un espace à trois dimensions et à 50 dimensions. Les divers niveaux d'abstraction du PGM sont représentés par des graphiques individuels empilés dans une même colonne de la figure. Chaque graphique représente les probabilités (en couleur) des divers états possibles (en ordonnée) en fonction de l'indice temporel (en abscisse). Une telle représentation permet de représenter l'état global du PGM étant donné une certaine séquence d'observations et de faire une comparaison avec d'autres PGMs en tenant compte de la topologie.

Les deux figures illustrent un exemple des résultats obtenus à partir d'une séquence de test réalisée selon le patron (4-1-2-3-4). Il est important de remarquer que le premier niveau du HHMM a deux états possibles (quatre dans le cas à 50 dimensions) dont un permet de reconnaître les primitives 1-2-4 du second niveau et l'autre, la primitive 3. Ainsi, le HHMM permet de reconnaître les états faisant partie d'un sous-patron. Afin de donner plus de

poids à ces observations sans alourdir inutilement le corps du texte, les figures illustrant les résultats obtenus pour les quatre autres patrons sont fournies à l'annexe 4 (espace à trois dimensions seulement). Les résultats relatifs aux observations à 50 dimensions sont pratiquement identiques.

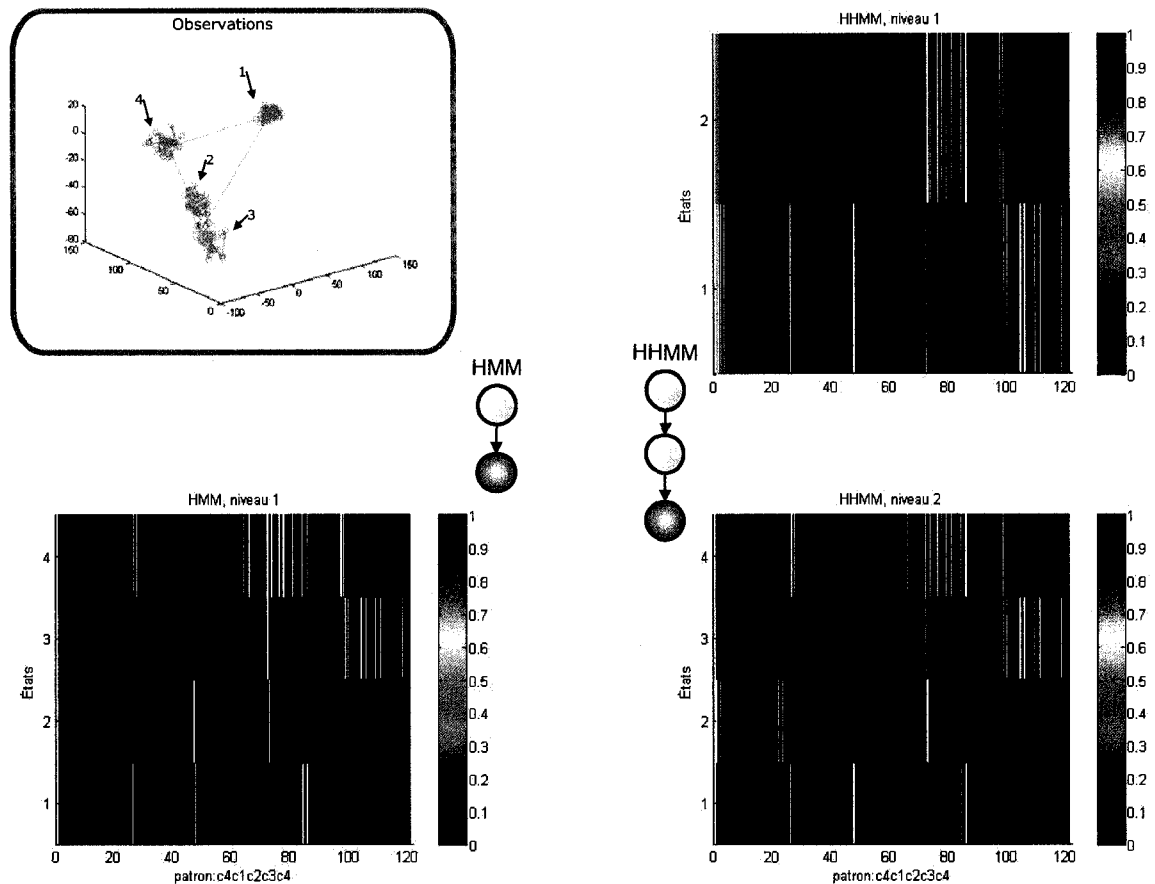


Figure 6.1 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à 3 dimensions. Les colonnes représentent les divers niveaux d'un même modèle. Chaque graphique représente un niveau de la topologie du modèle marginalisée (i.e. « figée ») étant donné une certaine séquence d'observations (4-1-2-3-4). L'indice temporel est en abscisse, les états possibles, en ordonnée et les probabilités sont indiquées par les couleurs.

6.1.2 Séquences (aléatoires) de primitives 2D

La démarche est analogue à celle suivie à la section précédente à la différence des observations qui sont d'une autre nature (primitives de type trajectoire bidimensionnelle) et du prétraitement en haute dimension qui est ajouté.

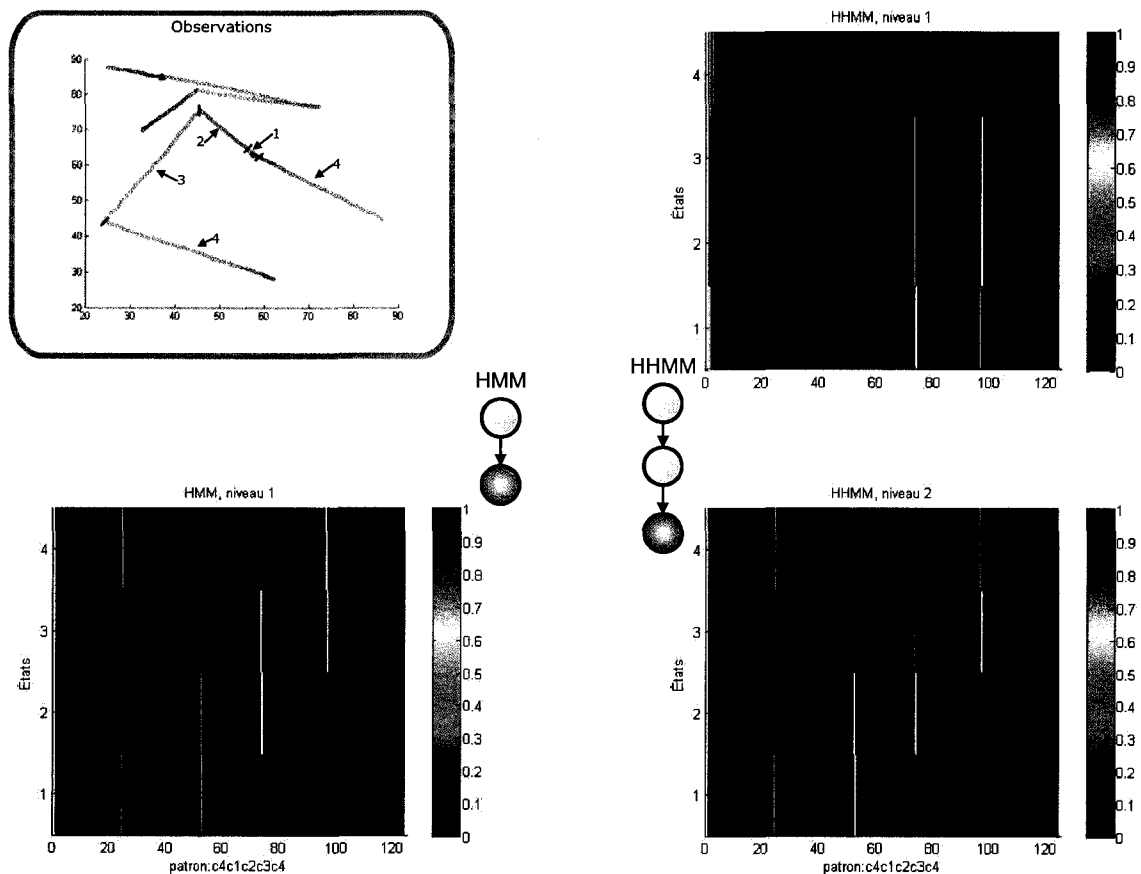


Figure 6.2 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à 50 dimensions. Les colonnes représentent les divers niveaux d'un même modèle. Chaque graphe représente la topologie du modèle marginalisée (i.e. « figée ») étant donné une certaine séquence d'observations (4-1-2-3-4). L'indice temporel est en abscisse, les états possibles, en ordonnée et les probabilités sont indiquées par les couleurs.

Le prétraitement en haute dimension considère une vitesse de déplacement maximale de 7 m/s et une fréquence d'échantillonnage de 5 Hz de sorte que le déplacement maximal à chaque échantillon devrait être de 1,4 m (hyper-paramètre `maxdist`). Afin de compartimenter la grandeur des déplacements en un histogramme, un filtre sur le déplacement minimal (hyper-paramètre `minres`) de 0,2 m est introduit. Ainsi, un histogramme sur les déplacements contient $2 \times \text{maxdist} / \text{minres} = 14$ compartiments. Une fenêtre temporelle (hyper-paramètre `winwidth`) de 10 échantillons (~ 2 s) est considérée pour « construire » l'histogramme. L'histogramme n'est pas normalisé (hyper-paramètre `normalize`). Les positions bidimensionnelles absolues (x, y) ne sont pas incluses dans la définition (hyper-paramètre `incOriginals`). L'histogramme sur les directions est toujours divisé en 9 compartiments, c'est-à-dire en huitième de cercle plus une valeur non orientée. Le tout ajouté du signal de statut pour un total de 52 dimensions. En somme, les hyper-paramètres du prétraitement en haute dimension sont :

<code>maxdist</code>	<code>minres</code>	<code>winwidth</code>	<code>incOriginals</code>	<code>normalize</code>	<code>Nsig</code>
1,4	0,2	10	0	0	52

De plus, il est important de noter que les PGMs sont plus complexes (en terme de degrés de liberté) que ceux de la section précédente puisque le nombre d'états du dernier niveau est passé à sept (pour les sept primitives de type trajectoire). Comme cela est représenté à la Figure 6.3, de tels modèles sont plus enclins au sur-apprentissage. Il est toutefois intéressant de remarquer que dans le cas du HHMM, le premier niveau tend à compenser le fait qu'il y ait sur-apprentissage au dernier niveau (modèle d'observation). La Figure 6.3 montre bien que le premier niveau reconnaît les primitives alors que les probabilités du dernier niveau sont éclatées sur divers états. De plus, les deux états du premier niveau ne suffisent pas à

discerner toutes les primitives comme le révèle la primitive 4 pour laquelle les probabilités sont partagées entre les deux états.

Tableau 6.3 : Scores des deux PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace à 52 dimensions (primitives de type trajectoire).

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)
1-(4-1-2)	501	88	83
(4-1-2)-7-4	609	72	77
2-4-7-1	468	74	75
7-(4-1-2)-4	614	83	75
(4-1-2)	344	82	66
	Score moyen	80	75

En somme, ces observations motivent le choix suivant quant à la topologie des PGMs : 7 états au dernier niveau, 4 états au premier niveau (HHMM seulement) et un *a priori* plus important sur la matrice de covariance afin de régulariser le modèle d'observation gaussien conditionnel.

Les scores des deux PGMs moyennés sur les 25 séquences de test dont les observations ont été prétraitées dans un espace à 52 dimensions sont compilés dans le Tableau 6.3. On y remarque que les deux PGMs sont comparables et que les scores ont diminués d'une vingtaine de pourcents par rapport aux résultats de la section précédente. Le HMM semble avoir un léger avantage sur le HHMM de l'ordre de 5%.

Les résultats relatifs aux PGMs employés seuls ont été présentés dans les deux contextes d'application où les primitives sont générées connues. Afin de faciliter la comparaison avec le PGM utilisé conjointement avec le modèle de segmentation en-ligne, les résultats du contexte d'application dans le cadre d'une réelle compétition *RoboCup* de même que les résultats relatifs

au coût de calcul en temps réel sont présentés aux sections 7.2.3 et 7.2.4 du prochain chapitre.

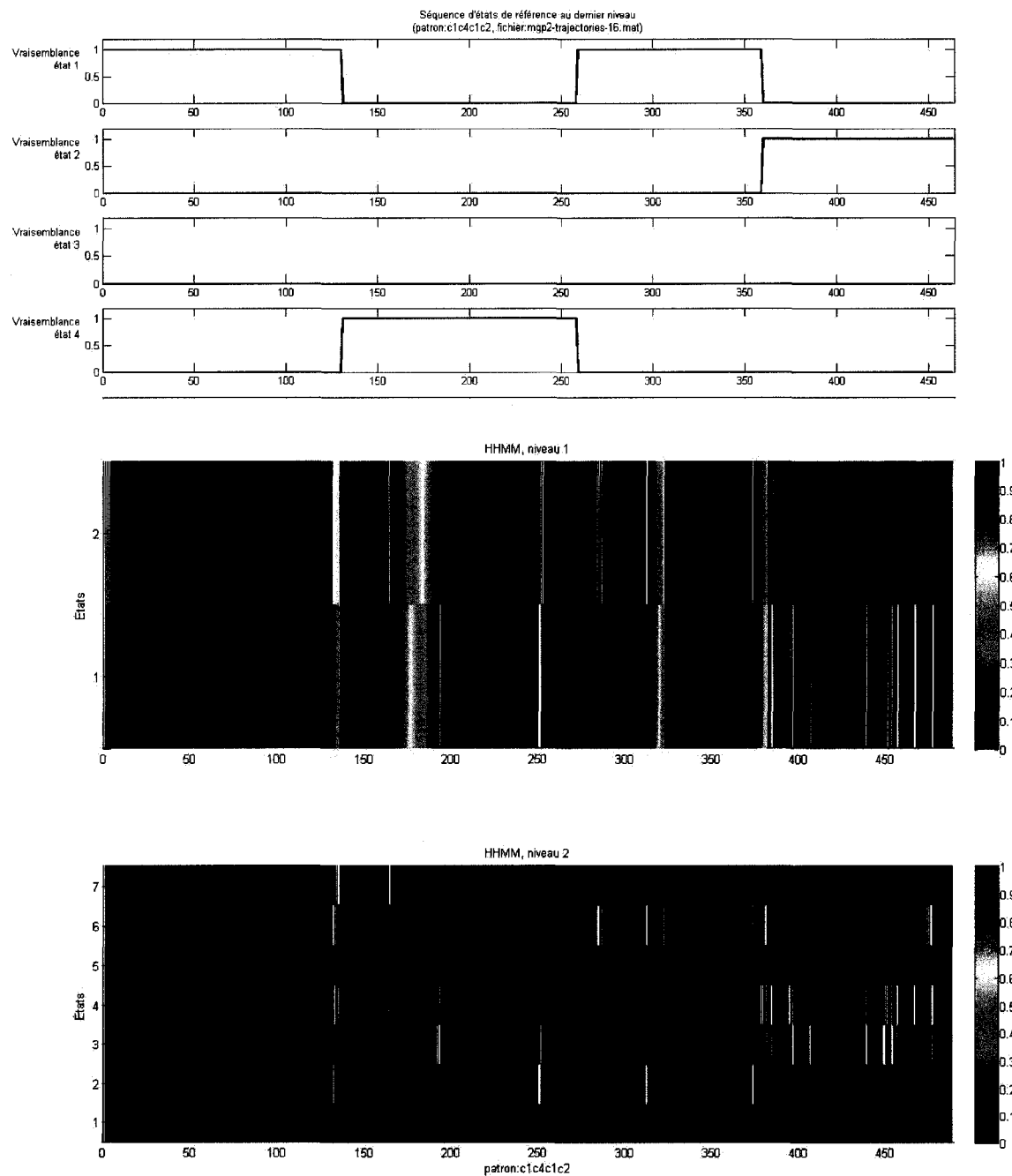


Figure 6.3 : Sur-apprentissage du modèle d'observation d'un HHMM et compensation du premier niveau (en bas). Les primitives d'origine sont présentées à titre de référence (en haut).

CHAPITRE 7 : « LA 2^{IÈME} DEMIE »

Les résultats des diverses expérimentations impliquant le PGM utilisé conjointement avec le modèle de segmentation en-ligne sont présentés dans ce chapitre. À l'instar du chapitre précédent, la validation du PGM est effectuée dans les trois contextes d'application introduits à la section 5.1. Le coût de calcul en temps réel est aussi présenté et comparé aux résultats obtenus jusqu'ici.

Il convient de rappeler ici que l'emploi du PGM avec le modèle de segmentation en-ligne implique un modèle d'observation discret permettant de traiter les primitives préalablement classifiées à partir du flux d'échantillons par un classificateur bayésien naïf. Le PGM traite donc de nouvelles observations uniquement lors d'un changement de primitive, diminuant de beaucoup le nombre d'inférences nécessaires. Les paramètres des CPTs représentent les probabilités de transition (dans le sens de diagramme de transition et non de transition temporelle markovienne) entre les divers états possibles. Au dernier niveau, ces états symbolisent les prototypes possibles de primitive.

Avant de procéder à la validation du PGM de façon analogue à la méthode employée au chapitre précédent, il est nécessaire de valider le modèle de segmentation en-ligne et de fixer la valeur optimale de ses hyperparamètres en vue d'une utilisation conjointe avec le PGM.

7.1 Validation du modèle de segmentation en-ligne

La méthodologie de la section 5.3.1 (A) est ici appliquée. Les expressions (5.1), (5.2), (5.3) et (5.4) sont utilisées afin de compiler les divers résultats de segmentation dans les tableaux. Quelques représentations graphiques sont utilisées pour supporter certaines spécificités.

7.1.1 Séquences (abstraites) de primitives gaussiennes

Tout d'abord, il est important de mentionner que les divers modèles testés ici se distinguent par la technique utilisée et par la valeur des hyper-paramètres. Les modèles sont identifiés par un nom (*e.g.* PPCA –11) dont la définition complète en terme des hyper-paramètres est fournie à l'annexe 2. Les hyper-paramètres font référence aux diverses quantités introduites à la section 4.4. Les quantités considérées pour évaluer le rendement sont moyennées sur dix (10) séquences de quatre (4) primitives gaussiennes.

Le Tableau 7.1 contient les taux et les scores de rendement des divers modèles de segmentation considérés dans un espace à 70 dimensions. Les modèles avec le plus haut rendement, c'est-à-dire ceux avec un score près de l'unité et un pourcentage de primitives détectées près de 100%, sont indiqués en vert. Ces modèles sont de type PPCA.

Les rendements les plus élevés offerts par les modèles de type PCA sont indiqués en jaune et montrent un score près de la demi-unité et un pourcentage de primitives détectées autour de 50%. Il est intéressant de relever que ces rendements sont présents uniquement dans la portion droite du tableau, soit dans le cas où le nombre de composantes principales est variable d'une primitive à l'autre. Cet aspect n'affecte toutefois pas les modèles de type PPCA.

L'éloignement des primitives entre elles n'affecte pas le rendement dans la mesure où la distance reste supérieure à un écart-type.

Le Tableau 7.2 contient des résultats similaires dans un espace à 25 dimensions. Seul le cas avec un nombre variable de composantes principales est présenté. La Figure 7.1 représente une trame de coupures permettant de comparer visuellement les accomplissements des modèles de segmentation des Tableau 7.1 et Tableau 7.2 sur une séquence typique de quatre primitives gaussiennes.

Tableau 7.1 : Rendement de divers modèles de segmentation en-ligne et effet du nombre de composantes principales effectives dans un espace à 70 dimensions en terme du taux de détection (R_c), du taux de succès (R_s), du taux de primitives détectées ($R_c R_s$) et du score.

Modèles testés	# composantes princ. similaire				# composantes princ. variable			
	R_c	R_s	$R_c R_s$	score	R_c	R_s	$R_c R_s$	score
	(%)	(%)	(%)	(0; 1)	(%)	(%)	(%)	(0; 1)
PCA dim-101	25,0	10,0	2,5	0,1	25,0	20,0	5,0	0,2
PCA dim-108	25,0	20,0	5,0	0,2	42,5	95,0	40,0	0,6
PCA dim-114	25,0	20,0	5,0	0,2	37,5	95,0	35,0	0,5
PCA dim-124	25,0	10,0	2,5	0,1	55,0	78,3	42,5	0,6
PCA dim-30	27,5	15,0	5,0	0,2	25,0	20,0	5,0	0,2
PCA dim-33	25,0	10,0	2,5	0,1	27,5	35,0	10,0	0,3
PCA dim-34	27,5	15,0	5,0	0,2	27,5	25,0	7,5	0,3
PCA vec-114	115,0	19,4	25,0	0,3	172,5	35,5	62,5	0,6
PCA vec-38	97,5	19,5	17,5	0,3	110,0	20,0	22,5	0,3
PCA vec-52	85,0	25,0	25,0	0,4	115,0	30,8	37,5	0,5
PCA vec-83	105,0	25,8	30,0	0,4	132,5	39,2	52,5	0,6
PCA vec-88	95,0	28,3	27,5	0,4	120,0	30,5	40,0	0,5
PCA vec-97	90,0	27,5	25,0	0,4	115,0	32,2	40,0	0,5
PPCA-1	92,5	45,8	42,5	0,6	82,5	49,2	42,5	0,6
PPCA-11	100,0	100,0	100,0	1,0	100,0	100,0	100,0	1,0
PPCA-24	100,0	87,5	87,5	0,9	100,0	90,0	90,0	0,9
PPCA-31	100,0	100,0	100,0	1,0	100,0	100,0	100,0	1,0
PPCA-46	100,0	100,0	100,0	1,0	100,0	100,0	100,0	1,0
Intervalle fixe : 50	297,5	14,3	22,5	0,2	182,5	12,5	22,5	0,2
Intervalle fixe : 240	50,0	5,0	2,5	0,1	52,5	8,3	5,0	0,1

Tableau 7.2 : Rendement de divers modèles de segmentation en-ligne dans un espace à 25 dimensions en terme du taux de détection (R_c), du taux de succès (R_s), du taux de primitives détectées ($R_c R_s$) et du score.

Modèles testés	# composantes princ. variable			
	R_c (%)	R_s (%)	$R_c R_s$ (%)	score (0; 1)
PCA dim-101	27,5	20,0	7,5	0,2
PCA dim-108	27,5	45,0	12,5	0,3
PCA dim-114	27,5	35,0	10,0	0,3
PCA dim-124	45,0	51,7	30,0	0,5
PCA dim-30	30,0	40,0	17,5	0,3
PCA dim-33	27,5	10,0	2,5	0,1
PCA dim-34	32,5	80,0	30,0	0,5
PCA vec-114	112,5	14,6	20,0	0,3
PCA vec-38	120,0	17,9	22,5	0,3
PCA vec-52	117,5	32,8	40,0	0,5
PCA vec-83	135,0	34,2	47,5	0,5
PCA vec-88	110,0	27,7	32,5	0,4
PCA vec-97	105,0	28,5	32,5	0,4
PPCA-1	65,0	71,7	45,0	0,7
PPCA-11	100,0	100,0	100,0	1,0
PPCA-24	97,5	81,7	80,0	0,9
PPCA-31	100,0	100,0	100,0	1,0
PPCA-46	100,0	100,0	100,0	1,0
Intervalle fixe : 50	175,0	8,5	15,0	0,2
Intervalle fixe : 240	47,5	10,0	5,0	0,2

Globalement, les modèles de type PCA avec le critère de dimensionnalité ont un faible taux de détection (25% à 50%) avec un taux de succès variable (10% et 95%). Les modèles de type PCA avec le critère sur la base vectorielle ont un taux de détection plus élevé que la normale (95% à 170%) compensant pour un faible taux de succès (15% à 40%). Les modèles de type PPCA ont, pour la plupart, des taux de détection et de succès près des optima (~100%).

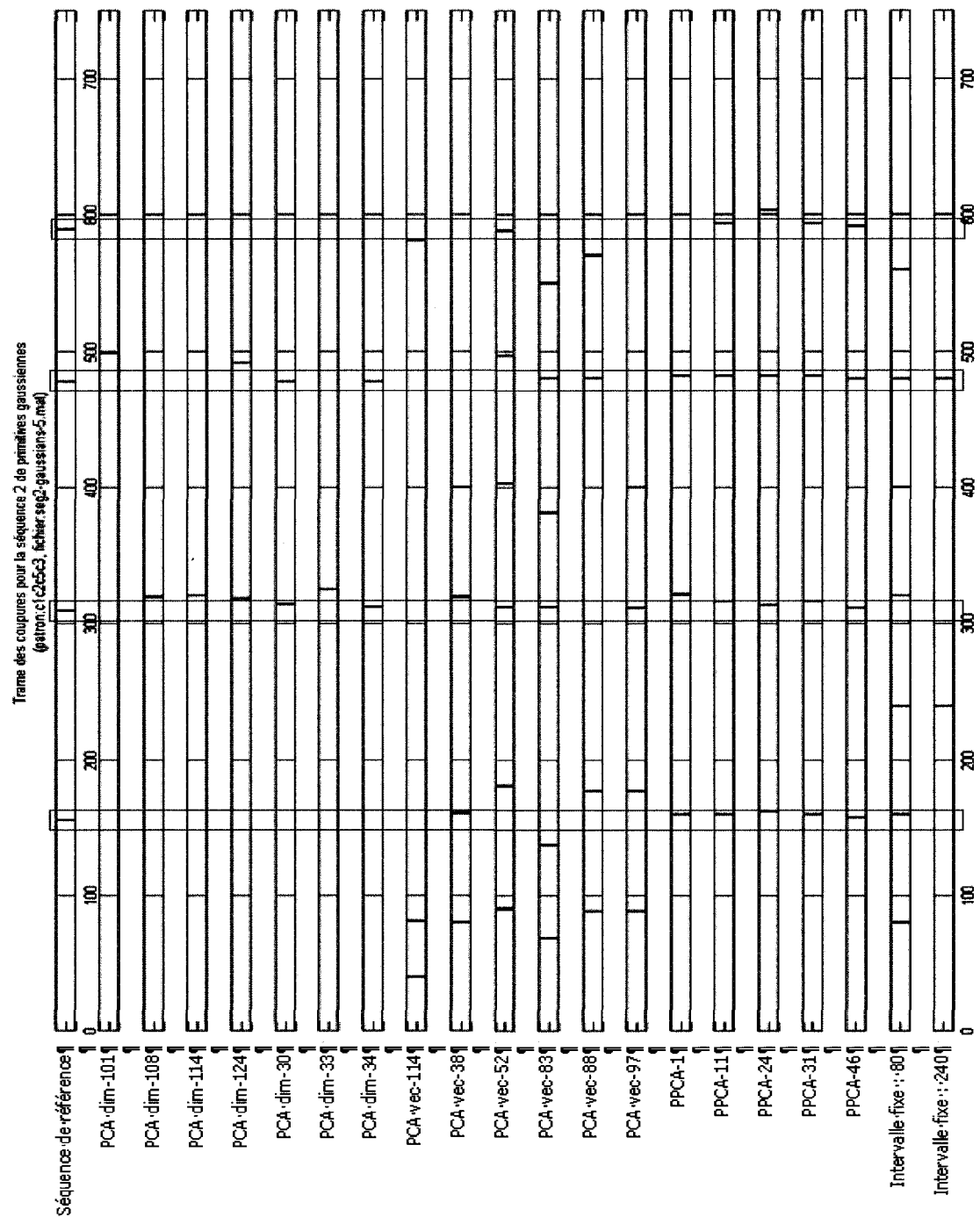


Figure 7.1 : Trame des coupures détectées par les divers modèles de segmentation en-ligne pour une séquence (1-2-5-3) de primitives gaussiennes en 70 dimensions.

7.1.2 Séquences (aléatoires) de primitives 2D

Les modèles testés dans cette section se résument aux modèles (ou à leurs variantes) présentant les meilleurs rendements à la section précédente. Plus spécifiquement, le modèle de type PCA avec le critère de dimensionnalité voit ses hyper-paramètres¹ fixés aux valeurs ci-dessous :

k (>Nsig)	l	m	maxlength	tol (%)	kstddev
35	5	1	500	0,99	2,6

alors que ceux du modèle de type PPCA sont fixés comme suit :

m (>Nsig)	inc (~0,1m)	maxlength	tol (%)	R
35	5	500	0,95	75

Le modèle de type PCA avec le critère de la base vectorielle est laissé pour compte étant donné le coût de calcul élevé. En effet, le calcul de l'expression (4.2) est plus coûteux que le calcul de l'expression (9.6), rendant le modèle inutilisable en temps réel.

Quant au prétraitement en haute dimension, les hyper-paramètres énoncés ci-dessous sont considérées : avec une largeur de fenêtre temporelle plus ou moins grande, avec ou sans les observations (x, y) , avec normalisation des histogrammes ou non.

maxdist	minres	winwidth	IncOriginals	normalize	Nsig
1,4	0,2	10	0	0	52

En définitive, les modèles dont la largeur de fenêtre temporelle est plus grande (*e.g.* 10 échantillons pour 2 s vs 4 échantillons pour moins d'une s)

¹ L'appellation des hyper-paramètres correspondre exactement à la formulation utilisée dans les divers modules Simulink de Matlab.

sont favorisés. Aussi, les modèles dont les histogrammes ne sont pas normalisés performant mieux que ceux avec les histogrammes normalisés. La présence des observations (x,y) n'influence pas les résultats outre mesure.

Afin de tester les limites des modèles de segmentation en-ligne, 21 patrons élémentaires sont sélectionnés. Chaque modèle est testé sur dix (10) réalisations distinctes de chaque patron élémentaire. Les sept types possibles de primitives bidimensionnelles introduits à la section 5.1.2 sont :

- 1- « immobile »
- 2- « rectiligne à vitesse constante »
- 3- « rectiligne avec accélération »
- 4- « rectiligne avec décélération »
- 5- « curviligne dans le sens horaire »
- 6- « curviligne dans le sens antihoraire »
- 7- « aléatoire »

Les patrons élémentaires sont :

2-1-2	3-1-4	4-1-3	5-1-6	2-1-6	4-1-5	7-1-7	7-2-7	7-3-7	7-5-7	3-2-4
5-2-6	5-3-6	5-4-6	4-3-4	5-6-5	1-7-1	2-7-2	4-7-3	6-7-5	2-7-3	

Le Tableau 7.3 contient le rendement des meilleurs modèles de type PCA et PPCA (*i.e.* ceux définis par les hyper-paramètres énoncés en début de section) moyenné sur les 210 séquences observées.

Tableau 7.3 : Rendement moyen des meilleurs modèles représentant les deux type de modèles de segmentation en-ligne considérés.

Modèles testés	R_c (%)	R_s (%)	$R_c R_s$ (%)	$score$ (0; 1)
PCA dim	67,0	52,9	40,8	0,6
PPCA	132,2	60,4	74,4	0,8
Intervalle fixe : 50	199,5	11,2	22,9	0,2
Intervalle fixe : 240	38,1	11,7	4,3	0,2

Les patrons pour lesquels le rendement est le plus bas sont ceux qui impliquent les primitives curvilignes (5, 6), surtout si la trajectoire bidimensionnelle ne révèle aucune boucle fermée (*i.e.* des tours incomplets comme le demi-tour ou le $\frac{3}{4}$ de tour), ainsi que les primitives accélérées/décélérées (3, 4). Dans ces situations, le taux de détection peut grimper jusqu'à 150-160% et le taux de succès, chuter jusqu'à 45%.

Comme le montrent les Figure 7.2 et Figure 7.3 dans le cas du modèle PPCA appliqué à une séquence 5-1-6, des coupures superflues sont introduites de façon importante pour les primitives 5 et 6.

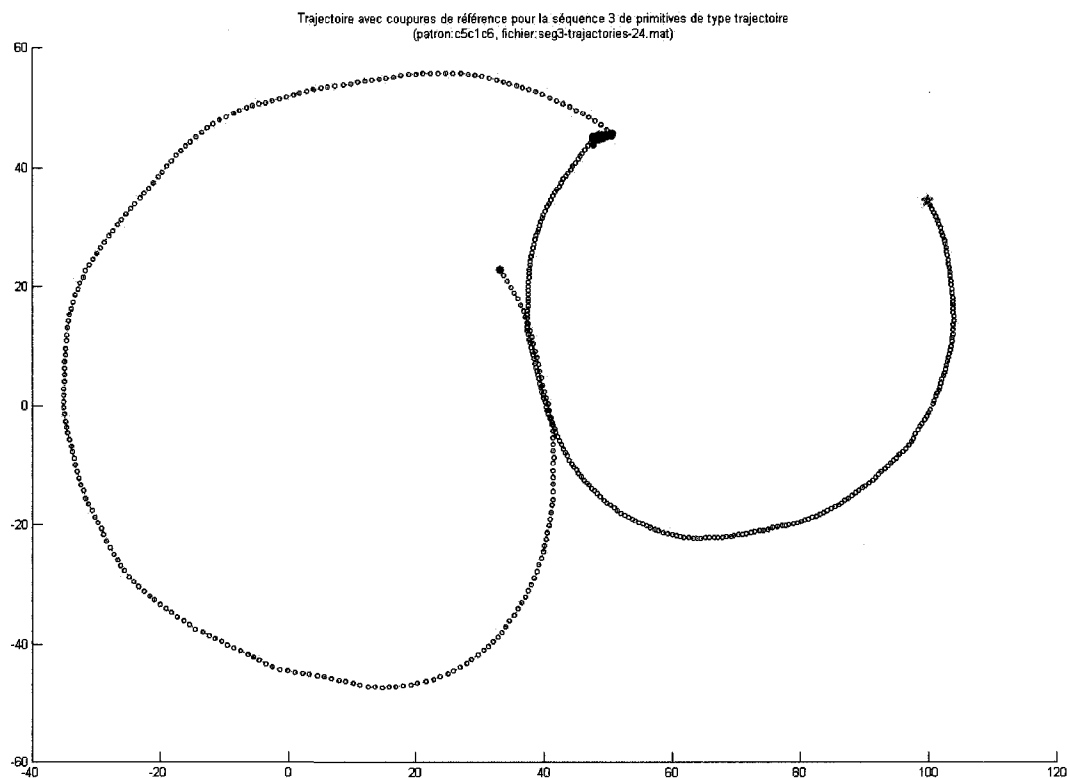


Figure 7.2 : Séquence de type trajectoire bidimensionnelle provenant du patron 5-1-6. Les points rouges marquent les limites des primitives. Le marqueur vert marque le début de la trajectoire.

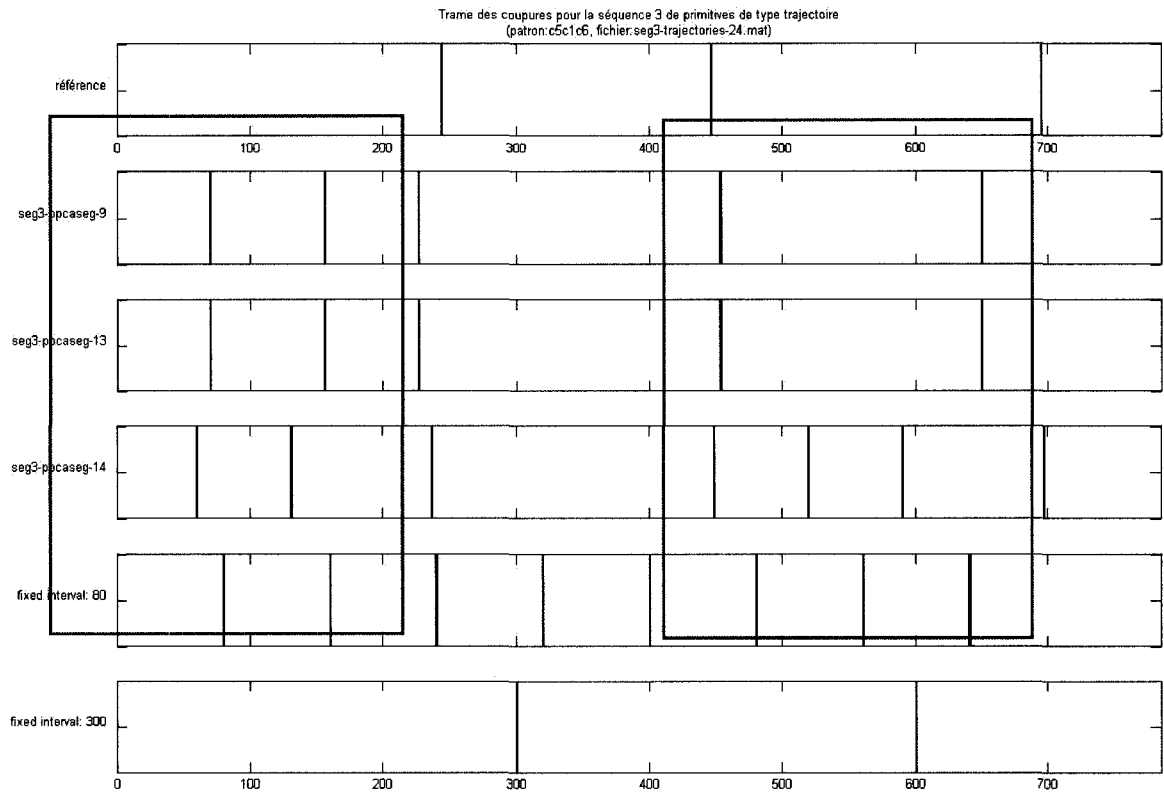


Figure 7.3 : Trame des coupures d’une séquence de type trajectoire bidimensionnelle provenant du patron 5-1-6.

7.1.3 Trajectoires réelles (RoboCup)

Le modèle de segmentation en-ligne considéré dans ce contexte d’application est le modèle de type PPCA présentant le plus haut rendement dans les résultats de la section précédente. Les trois trajectoires bidimensionnelles considérées pour effectuer les tests sont celles illustrées aux Figure 5.4, Figure 5.5 et Figure 5.6.

Le Tableau 7.4 présente quelques statistiques descriptives sur la dynamique des trajectoires bidimensionnelles considérée. Il est pertinent de remarquer que les vitesses atteintes par le ballon impliquent des déplacements dont la grandeur est supérieure (du double) à la valeur fixée par l’hyper-paramètre

« maxdist » du prétraitement en haute dimension. Ainsi, les observations rapportées dans cette section concernent davantage les deux autres agents mobiles.

Tableau 7.4 : Statistiques descriptives sur la dynamique des trois trajectoires bidimensionnelles considérées.

Figure	Agent mobile	Déplacement moyen par cycle (m)	Vitesse moyenne (m/s)	99% des déplacements entre (m)
23	joueur	0,45	4,5	(0; 1,2)
24	gardien	0,15	1,5	(0; 1,2)
25	ballon	1.05	10,5	(0; 2,8)

Un exemple des accomplissements du modèle de segmentation en-ligne PPCA est illustré à la Figure 7.4 où six primitives consécutives d'une portion de la trajectoire bidimensionnelle de l'agent mobile joueur sont représentées. La Figure 7.5 représente les traces sur les 1500 premiers échantillons de la distance de Mahalanobis moyenne des deux agents mobiles tel que décrit à la section 4.4.3.

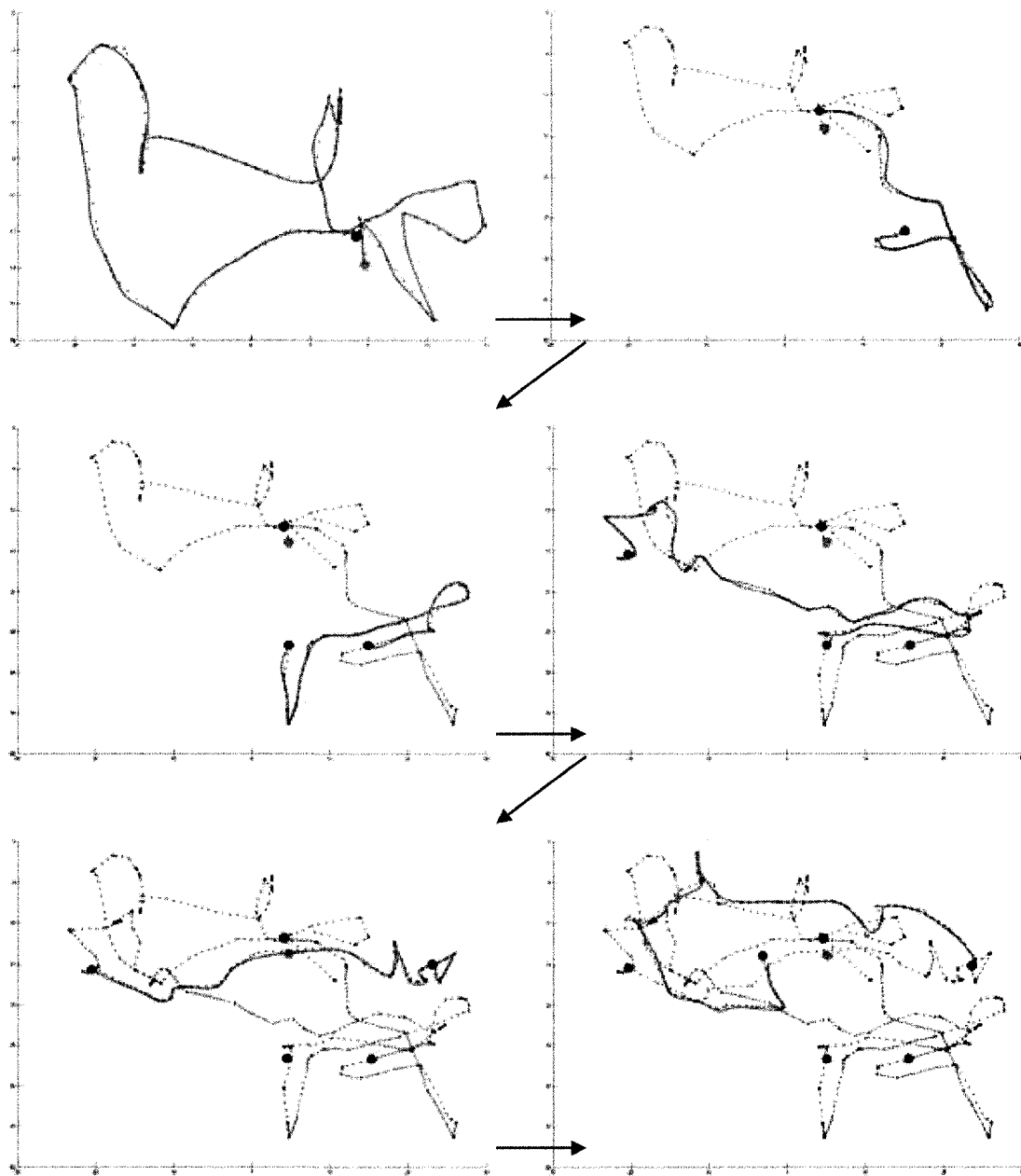


Figure 7.4 : Séquence de six primitives détectées à partir de la trajectoire bidimensionnelle de l'agent mobile joueur (échantillons 3425 à 4315 sur un total de 6000). Le point vert marque le premier échantillon alors que les points rouges marquent le commencement d'une nouvelle primitive.

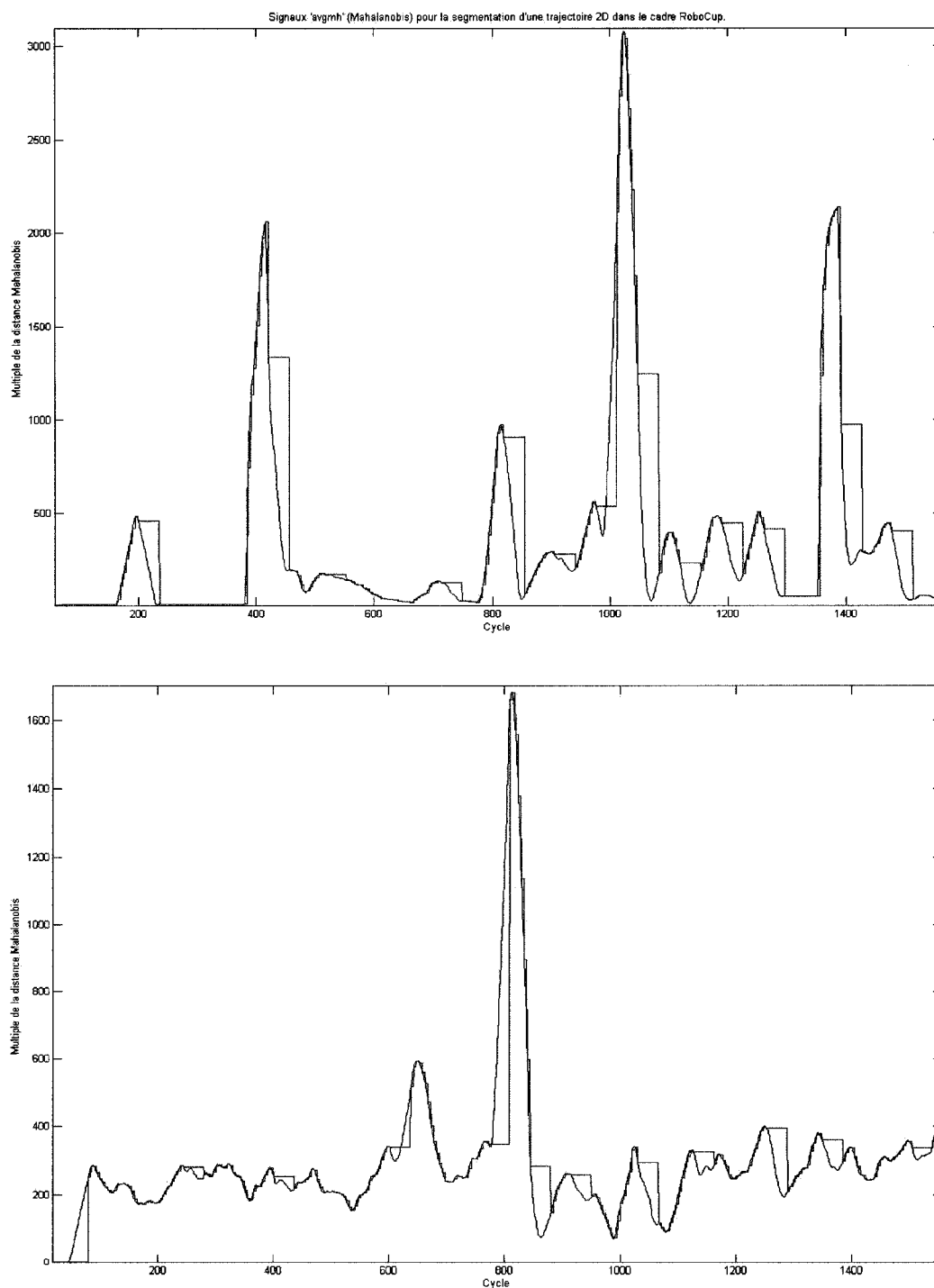


Figure 7.5 : Évolution de la distance de Mahalanobis moyenne pour les 1500 premiers échantillons de la trajectoire bidimensionnelle des agents mobiles gardien de but (en haut) et joueur (en bas). Les paliers marquent l'introduction d'une coupure.

7.1.4 Coût de calcul temps réel

Il a été mentionné à la section 7.1.2 que le modèle de segmentation en-ligne de type PCA avec le critère de la base vectorielle avait un coût de calcul plus élevé, le rendant du coup inutilisable en temps réel. Ce constat est ici formalisé à l'aide de la Figure 7.6 où le modèle PCA en question apparaît clairement 5 fois plus coûteux que le modèle de type PPCA et 10 fois plus coûteux que son homologue PCA basé sur le critère de la dimensionnalité.

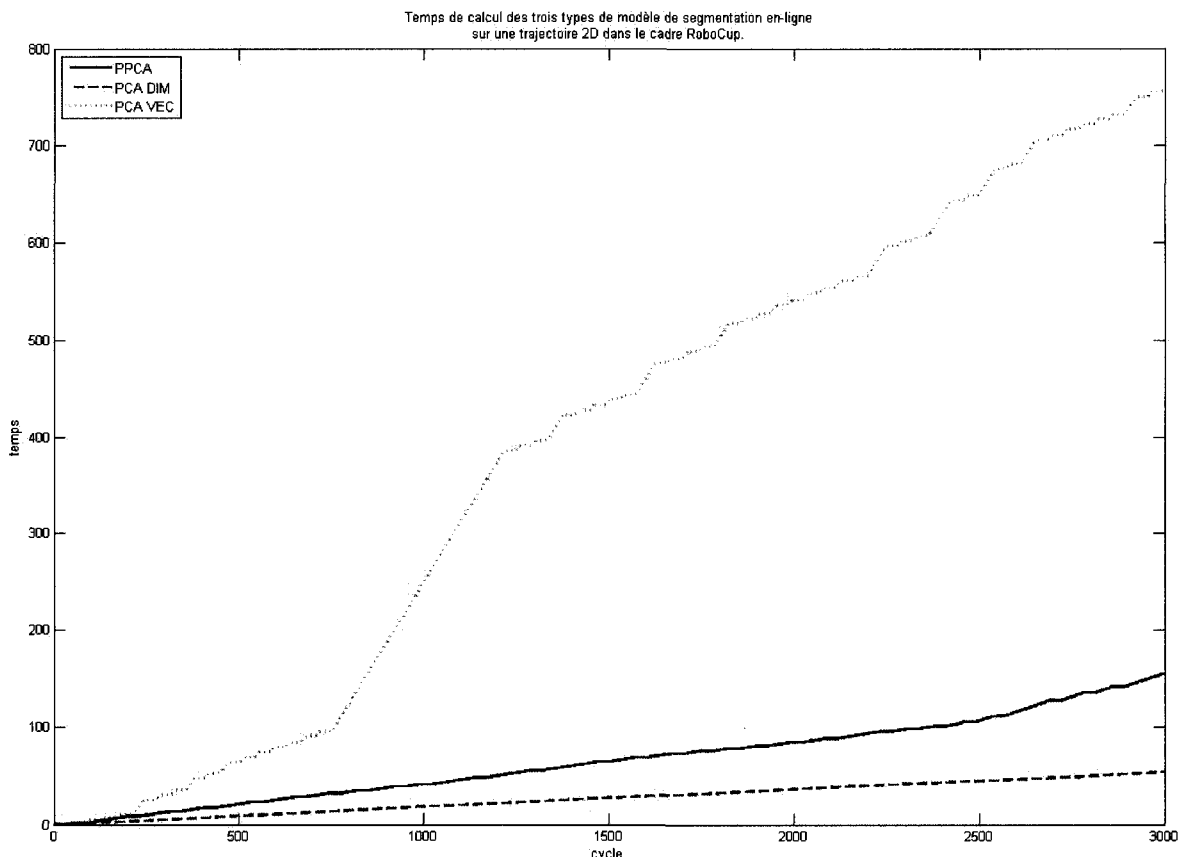


Figure 7.6 : Temps de calcul des trois types de modèle de segmentation en-ligne sur une trajectoire bidimensionnelle de 3000 échantillons. Le coût de calcul relatif au prétraitement en haute dimension est aussi considéré dans les résultats.

Le Tableau 7.5 résume les valeurs numériques des coûts de calcul des trois modèles de segmentation en-ligne. Le modèle PCA avec le critère de la base vectorielle dépasse la contrainte minimale pour les calculs en temps de plus du double. Certaines phases de calcul grimpent jusqu'à 0,6 s/cycle, une valeur 6 fois plus grande que le coût maximal attendu pour le modèle d'activité en entier. Le modèle PPCA consomme la moitié des ressources en temps alors que le modèle PCA avec le critère de la dimensionnalité se montre le moins gourmand avec une consommation de l'ordre du cinquième de la contrainte minimale.

Tableau 7.5 : Dérivée globale du temps de calcul correspondant à la pente de la courbe.

Modèles testés	Pente (s/cycle)
PCA dim	0,018
PPCA	0,051
PCA vec	0,252
Contrainte temps réel	0,1

7.2 Validation du PGM avec modèle de segmentation en-ligne

La méthodologie et le traitement des résultats menés dans cette section sont analogues à ce qui est fait la section 6.1 à la différence que le modèle de segmentation en-ligne de type PPCA est maintenant utilisé conjointement avec le HHMM. Tel que mentionné précédemment (*cf.* section 5.3), un classificateur bayésien naïf (NBC) est utilisé pour effectuer la classification des primitives de type trajectoire bidimensionnelle à l'entrée du HHMM.

7.2.1 Séquences (abstraites) de primitives gaussiennes

Les résultats obtenus à partir des 25 séquences sont compilés dans le Tableau 7.6 pour les primitives en trois dimensions et dans le Tableau 7.7 pour celles en 50 dimensions.

Tableau 7.6 : Scores des trois PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace de primitives gaussiennes à 3 dimensions.

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)	NBC+ HHMM (%)
1-(4-1-2)	102	100	100	100
(4-1-2)-3-4	127	100	99	100
2-4-3-1	100	99	100	100
3-(4-1-2)-4	122	100	100	100
(4-1-2)	71	99	99	100
	Score moyen	100	100	100

Les probabilités marginales dans le temps de chaque état de l'empilement du contexte des trois PGMs sont représentées à la Figure 7.7 (observations provenant d'un espace à 50 dimensions). Il est important de remarquer dans cet exemple que la séquence de primitives (3-4-1-2-4) est correctement reconnue par le NBC+HHMM, mais que les états du premier niveau sont moins discriminatifs que le HHMM quant au sous-patron (4-1-2). Le même phénomène (des probabilités dispersées au premier niveau) se produit pour les deux séquences dont les scores sont de 80% et 75%. Il est important aussi de rappeler que les HHMMs du Tableau 7.6 n'ont que deux états possibles au premier niveau et quatre au dernier alors que ceux du Tableau 7.7 en ont quatre au premier et sept au dernier.

Tableau 7.7 : Scores des trois PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace de primitives gaussiennes à 50 dimensions.

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)	NBC+ HHMM (%)
1-(4-1-2)	96	100	100	100
(4-1-2)-3-4	122	100	100	80
2-4-3-1	108	100	100	75
3-(4-1-2)-4	128	100	100	100
(4-1-2)	72	100	100	100
	Score moyen	100	100	91

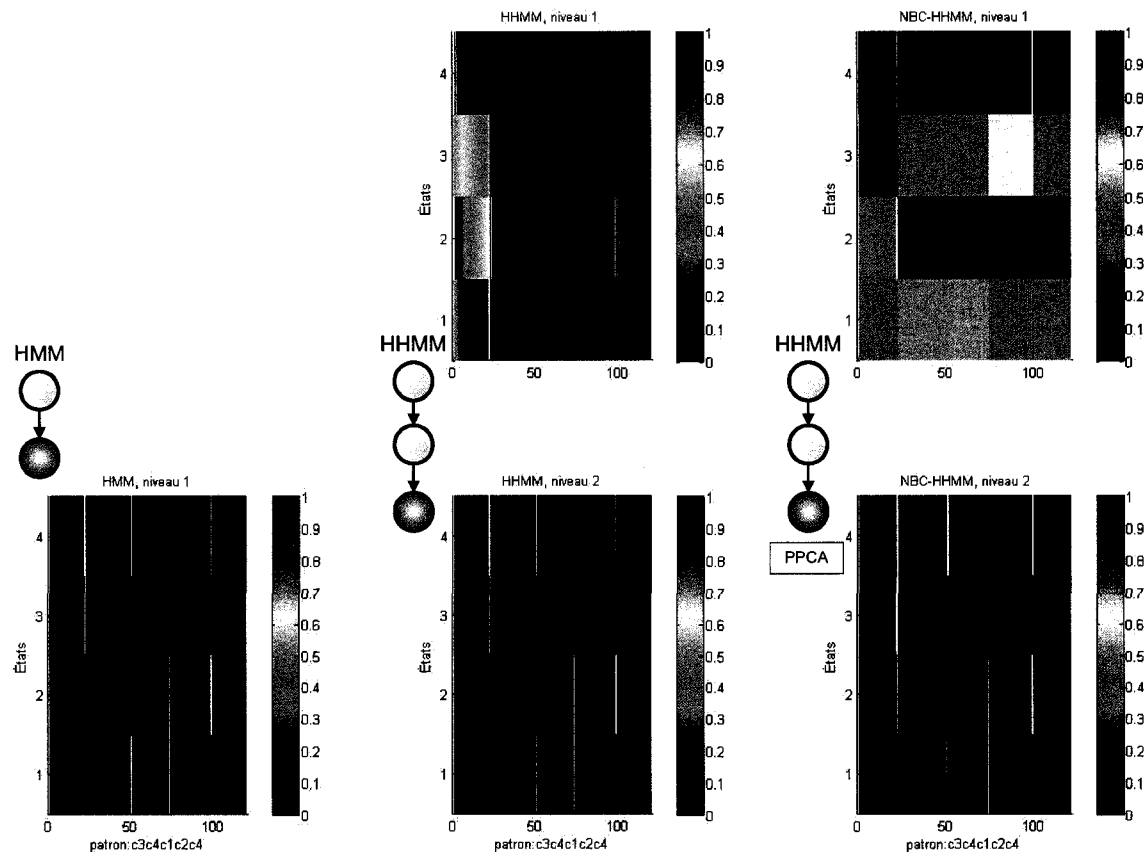


Figure 7.7 : Représentation de la probabilité marginale de chaque état dans l’empilement du contexte de chaque PGM pour des observations dans un espace de primitives gaussiennes à 50 dimensions. Les colonnes représentent les divers niveaux d’un même modèle. Chaque graphe représente la topologie du modèle marginalisée (i.e. « figée ») étant donné une certaine séquence d’observations (3-4-1-2-4). L’indice temporel est en abscisse, les états possibles, en ordonnée et les probabilités sont indiquées par les couleurs.

7.2.2 Séquences (aléatoires) de primitives 2D

Les scores des trois PGMs moyennés sur les 25 séquences de test dont les observations ont été prétraitées dans un espace à 52 dimensions sont compilés dans le Tableau 7.8. Les trois les scores du NBC+HHMM ont

diminués d'une trentaine de pourcents (comparativement à une vingtaine pour les deux autres modèles) par rapport aux résultats de la section précédente. Les scores du HHMM et du NBC+HHMM sont assez similaires pour conclure à une certaine forme « d'entendement » entre le modèle d'observation gaussien du HHMM et le duo NBC/modèle de segmentation en ligne.

Tableau 7.8 : Scores des trois PGMs, ou nombre de succès dans MPE / longueur de la séquence, dans un espace à 52 dimensions (primitives de type trajectoire).

Séquences de primitives	Nombre moyen d'échantillons	HMM (%)	HHMM (%)	NBC+HHMM (%)
1-(4-1-2)	501	88	83	67
(4-1-2)-7-4	609	72	77	60
2-4-7-1	468	74	75	58
7-(4-1-2)-4	614	83	75	87
(4-1-2)	344	82	66	89
	Score moyen	80	75	72

Les probabilités marginales dans le temps de chaque état de l'empilement du contexte des trois PGMs sont représentées à la Figure 7.8 (observations provenant d'un espace à 52 dimensions). Il est intéressant de remarquer que plusieurs états du dernier niveau sont associés à une même primitives étant donné l'existence de trois états superflus (les cinq patrons d'entraînement ne font usage que de quatre des sept primitives possibles).

Aussi, le premier niveau des deux HHMMs tend à placer les primitives 2 et 4 sous un même état, ce qui s'avère fort intéressant puisqu'il s'agit en effet de deux primitives rectilignes.

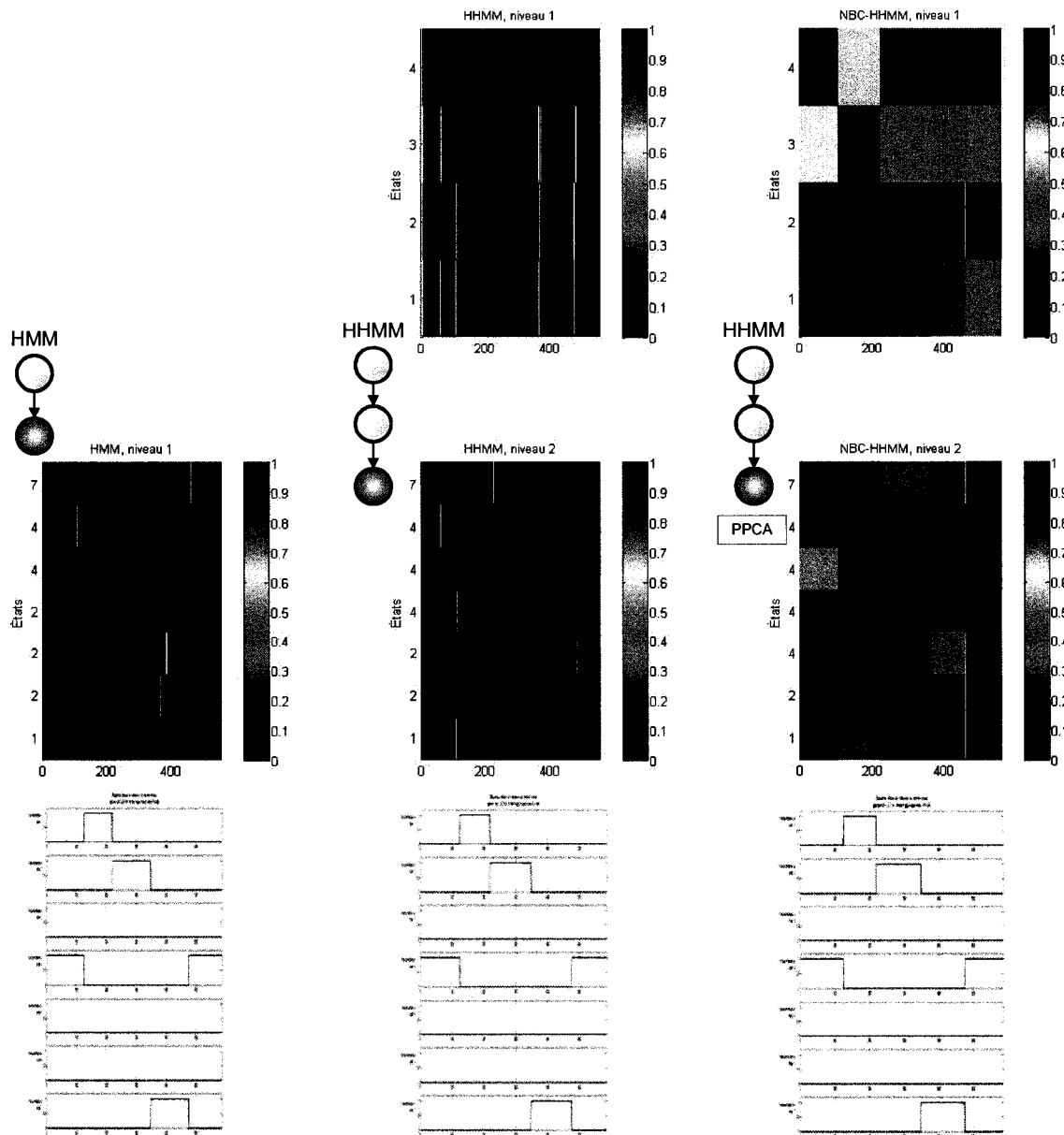


Figure 7.8 : Représentation de la probabilité marginale de chaque état dans l’empilement du contexte de chaque PGM pour des observations dans un espace de primitives à 52 dimensions obtenues à partir de trajectoires 2D générées. La marginale est calculée étant donné une certaine séquence d’observations (4-1-2-7-4) dont le patron est fourni au bas de la figure à titre de référence.

7.2.3 Trajectoires réelles (RoboCup)

Tel qu'énoncé dans la méthodologie décrite à la section 5.3.3, plusieurs portions de trajectoire d'environ 800 échantillons provenant des déplacements d'un joueur lors d'une même partie de soccer sont ici utilisées pour entraîner les divers PGMs. Les trajectoires bidimensionnelles sont préalablement traitées en haute dimension et segmentées en-ligne avec le modèle de type PPCA utilisé à la section précédente.

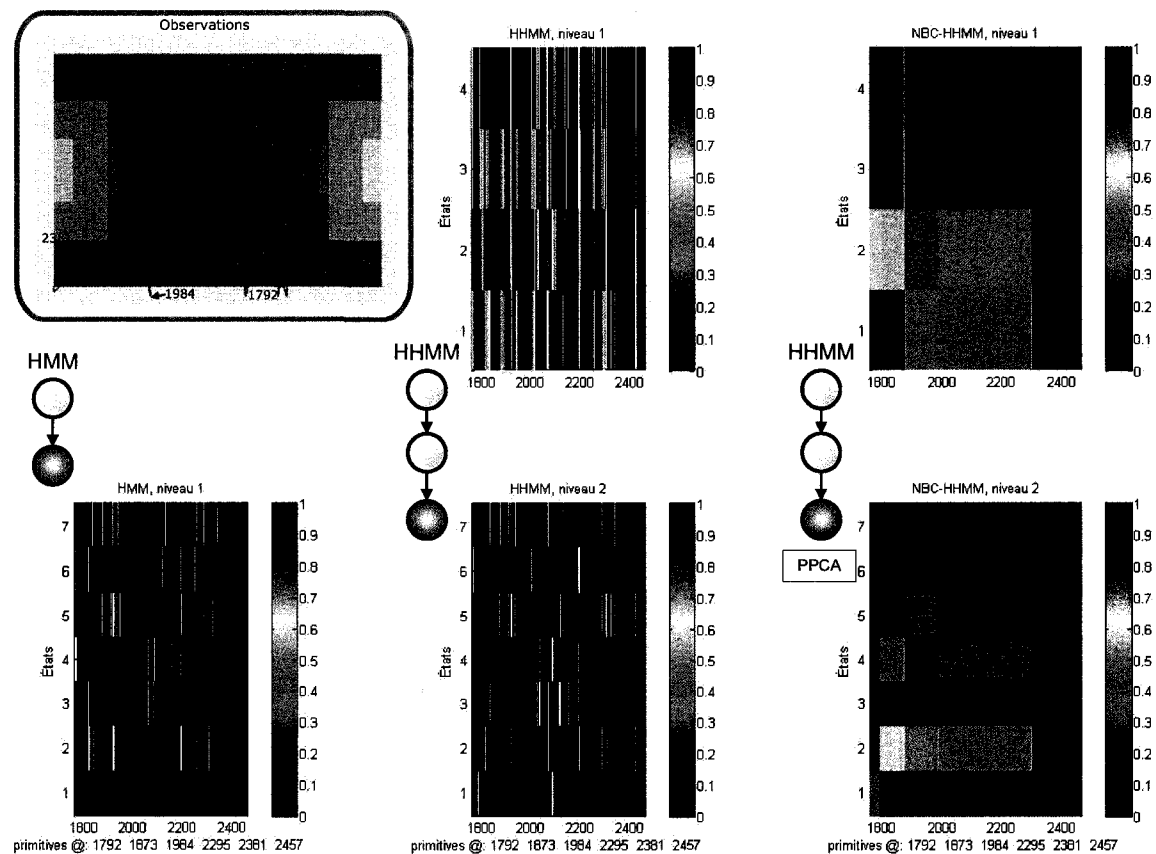


Figure 7.9 : Représentation de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM pour des observations dans un espace de primitives à 52 dimensions obtenues à partir d'une trajectoire bidimensionnelle observée dans une partie RoboCup. La portion de trajectoire considérée correspond aux échantillons 1759 à 2457. Les primitives détectées par le modèle de segmentation s'interfacent aux échantillons 1792, 1873, 1984, 2295 et 2381.

Une portion de trajectoire en milieu de partie est considérée en exemple pour l'agent mobile joueur. La Figure 7.9 représente les probabilités marginales dans le temps de chaque état de l'empilement du contexte des trois PGMs, toujours pour des observations provenant d'un espace à 52 dimensions.

La portion de trajectoire ici considérée contient six primitives détectées par le modèle de segmentation en-ligne. Les deux PGMs employés seuls montrent des états fragmentés indiquant des probabilités fortement éparpillées. Il est possible de remarquer au premier niveau du HHMM (employé seul) que certains états commencent à correspondre aux primitives détectées, mais toujours avec des probabilités fortement éparpillées. Il n'est toutefois pas possible d'avancer que le modèle d'observation gaussien du HHMM et le duo NBC/modèle de segmentation en-ligne parviennent aux mêmes conclusions comme c'était le cas à la section précédente.

Dans le cas du NBC+HHMM, les états du modèle sont beaucoup plus stables étant donné le traitement effectué par le modèle de segmentation en-ligne. L'exemple représenté à la Figure 7.10 montre que les primitives détectées par le duo NBC/modèle de segmentation en-ligne sont consistantes avec la réalité du jeu. À titre d'exemple, les deux primitives détectées entre les échantillons 2295 et 2457 et correspondant au changement d'état le plus franc au dernier niveau (2 vers 5) représentent en fait le passage de l'agent mobile joueur en touche en bordure du terrain.

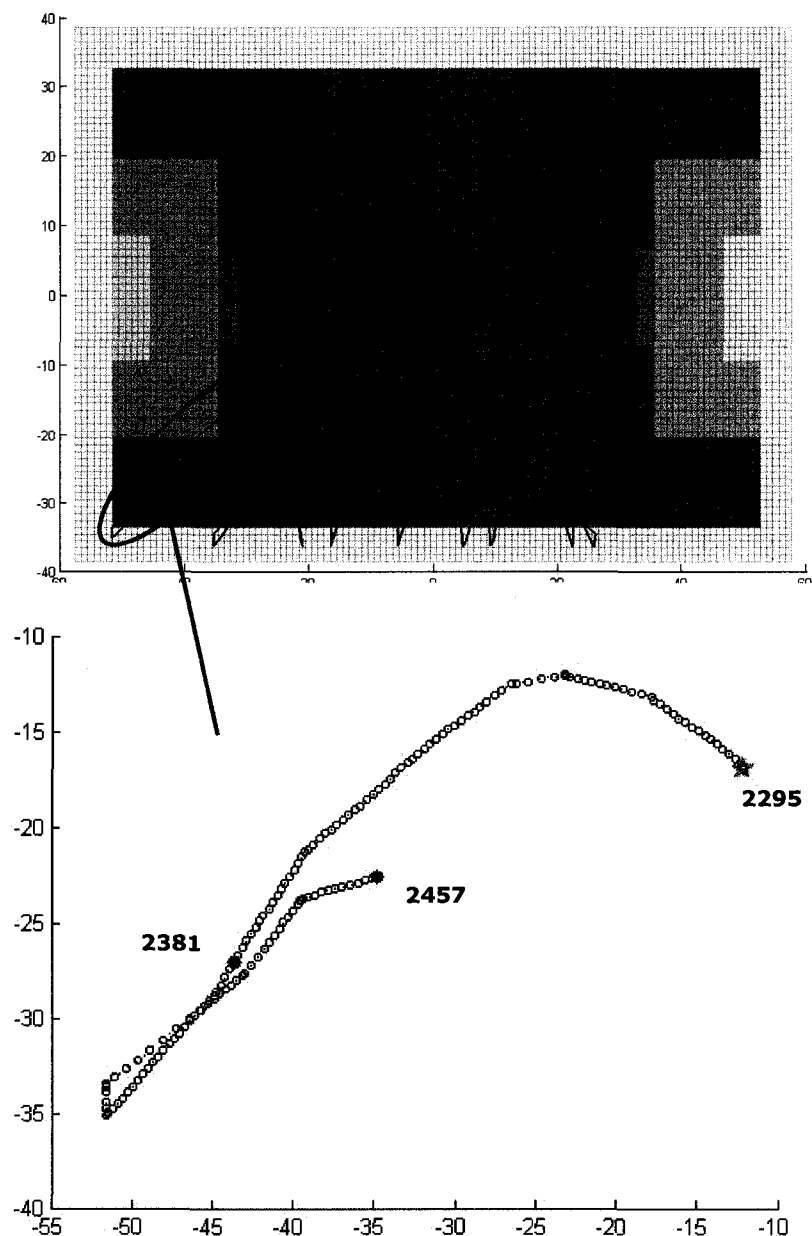


Figure 7.10 : (en haut).Trajectoire bidimensionnelle d'un joueur de soccer robotisé durant une partie complète d'une compétition RoboCup. (en bas) Deux primitives aisément détectées par le modèle d'activité représentant une pause (une touche) en bordure du terrain. Le marqueur vert marque le début de la première primitive (échantillon 2295) et les marqueurs rouges marquent la fin de chaque primitive (échantillons 2381 et 2457).

7.2.4 Coût de calcul temps réel

Les Figure 7.11 et Figure 7.12 représentent respectivement le coût de calcul du NBC+HHMM à deux niveaux (quatre et sept états) et du NBC+HHMM à trois niveaux (quatre, quatre et sept états) comparativement à trois autres PGMs : un HMM (sept états), un HHMM à deux niveaux (quatre et sept états) et un HHMM à trois niveaux (quatre, quatre et sept états). Le coût de calcul est représenté comme l'intervalle de temps écoulé pour le traitement d'un échantillon (dérivée instantanée de la courbe du temps cumulé).

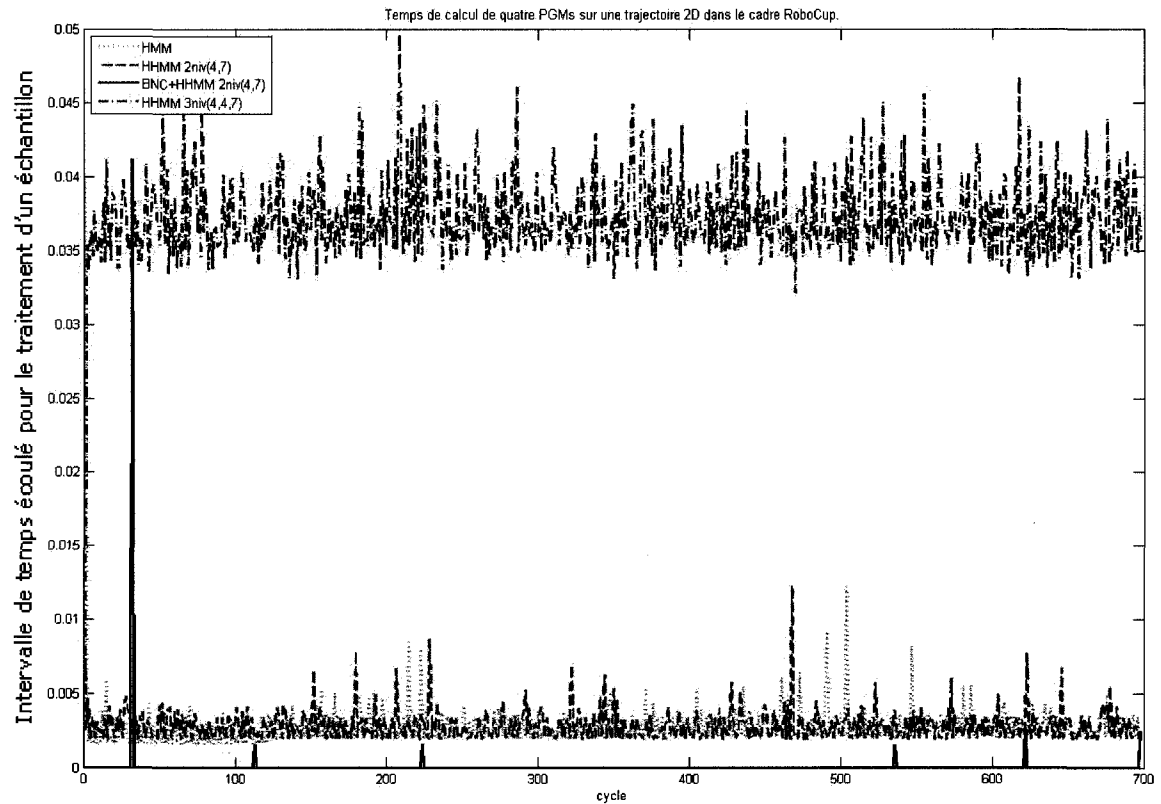


Figure 7.11: Temps de calcul de quatre PGMs sur une trajectoire bidimensionnelle de 700 échantillons représenté à partir de l'intervalle de temps écoulé à chaque échantillon. Le coût de calcul relatif au NBC n'est pas considéré dans le coût du HHMM avec modèle de segmentation en ligne.

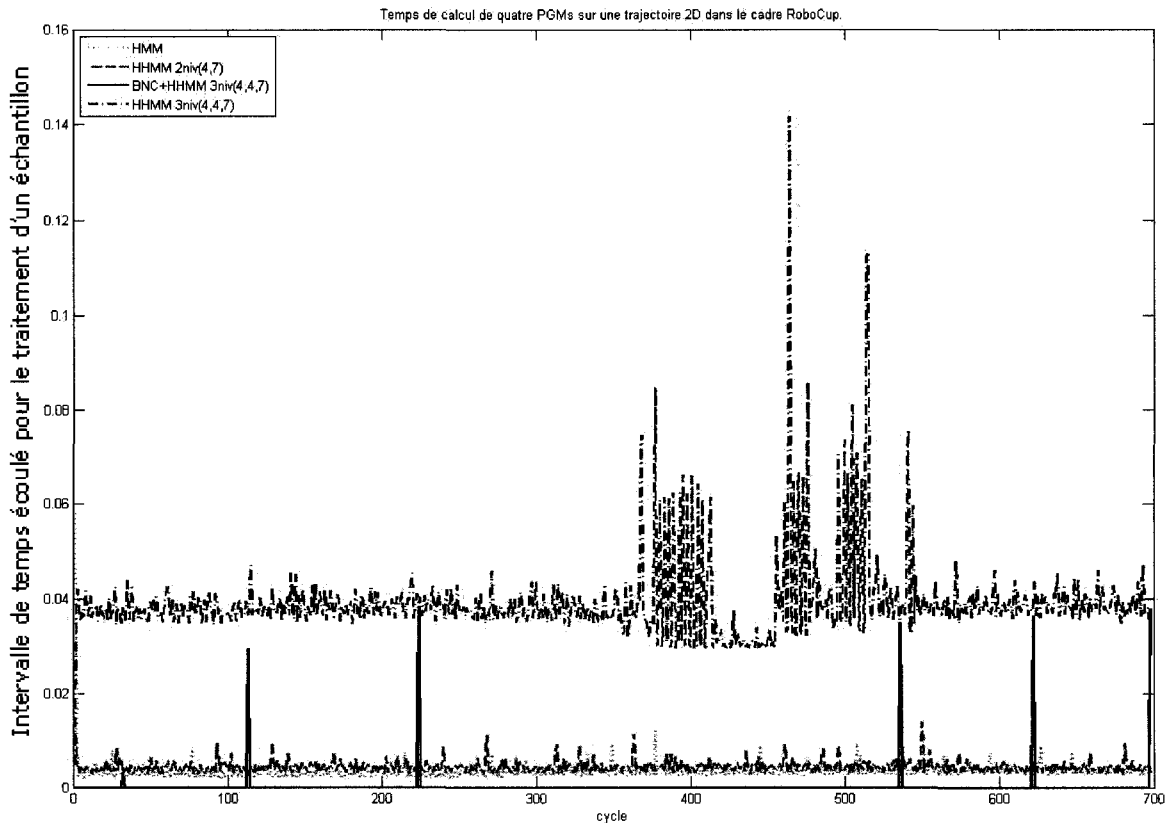


Figure 7.12 : Temps de calcul de quatre PGMs sur une trajectoire bidimensionnelle de 700 échantillons représenté à partir de l'intervalle de temps écoulé à chaque échantillon. Le coût de calcul relatif au NBC n'est pas considéré dans le coût du HHMM avec modèle de segmentation en ligne.

L'intervalle de temps écoulé pour le traitement d'un échantillon est le même pour le NBC+HHMM que pour le HHMM ayant une topologie équivalente. Cependant, le NBC+HHMM effectue le traitement uniquement lors d'un changement de primitive. Le Tableau 7.9 résume les valeurs moyennes et maximales typiques des coûts de calcul des PGMs considérés. Il est pertinent de remarquer que le coût de calcul est exponentiel selon le nombre de niveaux dans la hiérarchie. En effet, le HHMM à trois niveaux consomme la moitié des ressources temporelles alors que le HHMM à deux niveaux et le HMM en consomment respectivement environ 5% et 3%. La

complexité du HHMM limite donc rapidement le potentiel de traitement en temps réel du modèle d'activité. Le NBC+HHMM révèle un coût de calcul similaire à un HHMM doté d'une topologie équivalente, mais seulement momentanément de sorte que la valeur de l'intervalle moyen chute d'un facteur 10^2 .

Tableau 7.9 : Coûts de calcul (ou dérivées instantanées du temps de calcul) moyen et maximal typique de chaque PGM.

Modèles testés	Intervalle moyen (s/cycle)	Intervalle max typique (s/cycle)
HMM (7)	0,003	0,005
HHMM (4, 7)	0,004	0,005
HHMM (4, 4, 7)	0,04	0,05
NBC+HHMM (4, 7)	3×10^{-5}	0,004
NBC+HHMM (4, 4, 7)	3×10^{-4}	0,03
Segmentation PPCA	0,051	0,051
Contrainte temps réel	0,1	0,1

En somme, il a été montré dans ce chapitre que le modèle de segmentation en-ligne de type PPCA surclasse les modèles de type PCA en terme de rendement et de coût de calcul. Le HHMM utilisé conjointement avec ce modèle de segmentation en-ligne a été comparé à divers HMMs et HHMMs et montre des capacités de reconnaissance équivalentes aux autres PGMs, une aptitude à reconnaître des sous patrons, une meilleure capacité de généralisation dans le cas de trajectoires bidimensionnelles réelles et un coût de calcul sur l'ensemble d'une trajectoire inférieur au coût des autres PGMs. Ces résultats sont discutés avec plus de profondeur au prochain chapitre.

CHAPITRE 8 : « LES ECHOS DE VESTIAIRE »

À la lumière des résultats décrits dans les deux chapitres précédents, il est d'intérêt de faire un retour sur les contributions de cette recherche et sur le concept de modèle d'activité et d'en discuter davantage la nature, la capacité, les limites et le potentiel.

8.1 Les principales contributions

Les principales contributions de cette recherche sont les suivantes :

- Le concept de primitive gaussienne construite à partir de statistiques sur la dynamique de la trajectoire est introduit.
- Le modèle d'activité est défini en terme de ces primitives (et, par conséquent, de la dynamique de la trajectoire) plutôt que seulement en terme de positions de référence (*landmarks*) ou de positions sur une grille. L'intérêt quant aux déplacements d'un agent mobile se situe plutôt dans le « comment » plutôt que sur le « où » (quoiqu'il est aussi possible de faire les deux avec le modèle proposé).
- Le couplage du HHMM avec un modèle de segmentation en-ligne basé sur PPCA réduit la charge de calcul d'inférence et rend le modèle d'activité plus robuste face à la grande variabilité des trajectoires.
- Il s'agit à ma connaissance d'une première application du modèle HHMM au soccer robotisé.

8.2 Le rendement et les limites du PGM

Les traitements effectués sur les trois PGMs (HMM, HHMM et NBC+HHMM) dans les contextes d'application contrôlés, c'est-à-dire à partir de primitives générées dont l'identité est connue, ont permis de valider l'aptitude des trois PGMs à reconnaître des séquences quelconques de primitives et aussi, dans le cas des HHMMs, à reconnaître des sous patrons se produisant fréquemment. En général, les scores représentant la capacité à reconnaître une séquence de primitives à partir d'une séquence d'observations sont du

même ordre de grandeur pour les trois PGMs. Le score du HHMM est inférieur à celui du HMM de quelques pourcents alors que le score du NBC+HHMM est inférieur de 5 à 10%. Cet effet est d'autant plus apparent lorsque la dimensionnalité des observations est élevée.

Plus spécifiquement, les scores sont de l'ordre de 90 à 100% dans le cas idéal de primitives gaussiennes et de 70 à 80% dans le cas de primitives de type trajectoire. Puisque des dimensionnalités similaires ont été testées, la chute d'une vingtaine de pourcents dans le contexte de trajectoires bidimensionnelles s'explique par une plus grande variabilité des échantillons et par l'existence d'échantillons atypiques. Une régularisation des paramètres du modèle d'observation gaussien (*a priori* sur la matrice de covariance) s'avère nécessaire pour éviter le sur-apprentissage.

8.2.1 Le nombre de paramètres

La légère baisse de rendement dans le cas du HHMM est applicable au fait que ce PGM est d'une complexité plus grande que le HMM, c'est-à-dire que le nombre de paramètres augmente exponentiellement avec le nombre d'états dans la hiérarchie (s'il y a K états par niveau, il y a K^D états possibles au total et K^{2D} paramètres uniquement pour le modèle de transition temporelle). Dans le cas du NBC+HHMM, la situation se complique davantage puisqu'en faisant passer la nature de l'observation « d'échantillon d'une primitive » à « primitive complète », le nombre d'exemples d'apprentissage pour le MGP diminue proportionnellement à la longueur moyenne d'une primitive. Par exemple, dans le contexte d'application des primitives de type trajectoire bidimensionnelle, une primitive contient en moyenne 120 échantillons (paramètre du générateur) de sorte qu'un ensemble de 25 trajectoires d'entraînement contient environ 12k échantillons de diverses primitives ($25 \times 4 \times 120$) au lieu de 100 exemples de primitives (25×4). En somme, un modèle avec plus de paramètres et moins

d'exemples d'apprentissage dans un espace avec une grande dimensionnalité voit nécessairement son rendement diminuer.

8.2.2 Les primitives gaussiennes

Le fait que les trois PGMs aient des scores et des représentations similaires de leurs marginales (du moins au dernier niveau) indique que les primitives gaussiennes sont reconnues autant par le modèle de segmentation en-ligne PPCA (cas NBC+HHMM) que par le modèle d'observation gaussien du PGM (cas HMM et HHMM). Cette observation a d'autant plus de poids que les deux techniques sont fort différentes : la première considère la similarité d'une suite d'échantillons selon un modèle probabiliste multi-varié basé sur des exemples (*i.e.* paramètres non fixés ou *instance-based*) dans une espace de dimensionnalité réduite alors que la seconde considère un processus stochastique de prédiction/transition/mise à jour d'états à partir d'un modèle probabiliste d'observation multi-varié aux paramètres fixés.

Aussi, le fait que les HHMMs classent certaines primitives sous un même état plus abstrait (e.g. les primitives 2 et 4 de type trajectoire bidimensionnelle respectivement « rectiligne à vitesse constante » et « rectiligne avec décélération »), montre bien que le modèle d'activité est en mesure de regrouper des trajectoires ou des portions de trajectoires en fonction de leur dynamique.

Toutefois, la configuration des états abstraits est difficilement reproductible dans un cadre non supervisé et les résultats dépendent beaucoup de l'initialisation des paramètres dans EM.

8.2.3 Anomalies et traitement en lots

Dans le contexte d'application de trajectoires bidimensionnelles provenant de réelles compétitions RoboCup, les deux PGMs employés seuls sont grandement affectés par la variabilité des observations et par la présence d'échantillons atypiques (*outliers*). Ainsi, le processus stochastique

d'évolution des états est affecté par la présence d'anomalies dans le traitement séquentiel des observations. Le NBC+HHMM est plus robuste quant à l'existence d'anomalies puisque les observations sont traitées en lot et que les changements d'état surviennent uniquement lors d'un changement de primitives spécifié par le modèle de segmentation en-ligne.

8.2.4 Description non supervisée des activités

Un modèle d'activité basé sur le NBC+HHMM et un modèle de segmentation en-ligne de type PPCA parvient à capturer les particularités d'une trajectoire bidimensionnelle. À titre d'exemple, le NBC+HHMM à deux niveaux de la Figure 7.9 montre (après visionnement de la séquence de jeu) l'existence de trois phases dans la trajectoire : de 1759 à 1984 (état 2 jaune), il y a « départ en touche », de 1985 à 2295 (états 2 et 4 bleu-vert), deux « manœuvres sur le terrain » sont exécutées, de 2295 à 2457 (état 5 rouge), il y a « retour en touche » puis « remise en jeu ». Ces phases ne sont détectées uniquement qu'à partir de l'observation de la dynamique de la trajectoire et non par l'existence de quelconques règles de jeu.

8.2.5 Traitement en temps réel

Bien que les PGMs évalués dans ce travail aient tous été en mesure de traiter les observations à l'intérieur de la contrainte minimale de traitement en temps réel (0,1s), il demeure que la complexité de la topologie du HHMM limite rapidement (exponentiellement selon la profondeur de la hiérarchie) le potentiel de traitement en temps réel du modèle d'activité. Le NBC+HHMM est sujet aux mêmes contraintes puisque l'algorithme d'inférence (exact) reste le même.

Toutefois, le NBC+HHMM présente l'avantage d'un coût de calcul momentané, c'est-à-dire qu'une fois le traitement effectué, le modèle d'activité est considéré « à jour » et « en temps réel » jusqu'au prochain traitement. C'est-à-dire que l'état décrit par le HHMM est assuré valide (ou le plus probable) à l'instant même de l'observation puisque le modèle de segmentation en-ligne ne donne pas raison de croire autrement.

Un tel traitement pourrait être qualifié de « quasi temps réel ». Un traitement complètement en temps réel ne peut être obtenu qu'en réduisant le coût de calcul par l'utilisation d'algorithmes moins coûteux et potentiellement approximatifs.

8.3 Le rendement et les limites du modèle de segmentation en-ligne

Les résultats montrent clairement que le modèle de segmentation de type PPCA surclasse les modèles de type PCA avec un rendement 2 à 4 fois supérieur pour un coût de calcul fort acceptable. Dans le cas d'application de primitives gaussiennes, le rendement du modèle PPCA s'avère pratiquement sans faute et ce, sans égard à la dimensionnalité ou à la variabilité du nombre de composantes principales entre les primitives. Le taux de détection atteint près de 75% dans le cas de primitives de type trajectoire bidimensionnelle, un rendement équivalent au double de celui offert par le meilleur modèle de type PCA. Le score de 0,8 atteint par le modèle de type

PPCA indique, par opposition aux scores obtenus avec les modèles à intervalle fixe, un minimum de détections superflues et un maximum de détections valides.

La pertinence du modèle PPCA est d'autant plus grande qu'il s'inscrit dans les optiques « générative » et « temps réel » du modèle d'activité.

8.3.1 Robustesse face à la dimensionnalité

Le critère de dimensionnalité à lui seul ne suffit pas à détecter les primitives comme le montrent les résultats du modèle de type PCA avec le critère de la dimensionnalité. Ce modèle est grandement affecté par la variabilité du nombre de composantes principales de même que par la dimensionnalité initiale de l'espace des primitives. En effet, l'espace de plus petite dimensionnalité (25), ayant un nombre moindre de degrés de liberté, affecte à la baisse le rendement du modèle PCA. Le modèle PCA avec le critère de la base vectorielle introduit, quant à lui, un grand nombre de détections superflues en plus d'avoir le défaut de consommer énormément de ressources.

Il est important de remarquer ici que dans les trois algorithmes, la longueur initiale d'une primitive est de 35 ou 40 échantillons (hyper-paramètres k , l , m) alors que la dimensionnalité peut atteindre 50, voire même 70. Cet aspect peut se révéler problématique pour évaluer la matrice de covariance à la base de la décomposition en composantes principales, comme c'est le cas notamment dans le modèle de type PCA avec le critère de la base vectorielle. En effet, la base choisie (et fixée) est rarement valide et il en résulte un grand nombre de coupures de longueur minimale. Dans le cas de PCA avec le critère de la dimensionnalité, la première estimation du nombre de composantes principales est aussi souvent erronée pour la même raison.

Le modèle PPCA est toutefois plus robuste quant au nombre ou à l'identité des composantes principales puisque celles-ci sont toutes intégrées à une même matrice de covariance, soit explicitement soit sous forme de bruit.

Cette matrice est utilisée dans le calcul d'une statistique (la distance de Mahalanobis moyenne) en plus d'être mise à jour itérativement avec davantage de données. Cette façon de faire laisse place à une certaine marge d'erreur quant à l'évaluation des composantes principales dans la mesure où le lot d'échantillons utilisé pour le calcul de la statistique est susceptible d'être représenté par la matrice en question.

Advenant le cas où le modèle PPCA a de la difficulté à représenter, la valeur de la statistique reste élevée après l'introduction d'une coupure. Un tel cas est représenté à la Figure 7.5 par la trace de la distance de Mahalanobis moyenne des deux agents mobiles au sein d'une compétition RoboCup. Les primitives détectées pour le gardien de but sont beaucoup plus franches que celles détectées pour le joueur.

8.3.2 Limitation quant à la résolution

Il a été mentionné que les primitives de type trajectoire bidimensionnelle curvilignes (5, 6) et rectilignes accélérées/décélérées (3, 4) sont à l'origine des baisses de rendement observés dans le second contexte d'application. Des coupures superflues sont introduites de façon importante pour les primitives 5 et 6.

Ce résultat est une conséquence directe de la résolution de la fenêtre temporelle utilisée dans le prétraitement en haute dimension. En effet, une courbure quasi constante, c'est-à-dire qui ne se referme pas à l'échelle du modèle, a pour effet de faire varier progressivement l'histogramme des directions de sorte que les primitives dont la longueur est supérieure à trois fois l'hyper-paramètre m (*i.e.* longueur d'environ 100-110) présentent des différences significatives entre la fin et le début de la trajectoire. Le même phénomène s'applique à l'accélération et à l'histogramme sur la distance.

La Figure 8.1 illustre un exemple où une trajectoire curviligne (celle de la Figure 7.2) peut être perçue autrement à la résolution du modèle de segmentation (de l'ordre de 1s ou 10 échantillons). En effet, du point de vue

de cette résolution, les deux trajectoires issues de deux patrons distincts pourraient fort bien provenir du même patron.

En somme, la détection de primitives additionnelles dans de telles circonstances devraient être interprétée comme une simplification d'une première primitive (qui n'en est apparemment pas une) en plus petites primitives et non comme l'introduction de détections superflues. Ce phénomène devrait être compensé au niveau du HHMM moyennant l'ajout d'un niveau supplémentaire.

8.4 Avenues potentielles pour le modèle d'activité

Il existe plusieurs avenues prometteuses pouvant améliorer le travail amorcé.

Tout d'abord, les difficultés liées à l'apprentissage non supervisé et plus spécifiquement à la reproductibilité de la configuration des états abstraits pourraient être abordées en envisageant une topologie complètement connectée telle que représentée à la Figure 4.7 ou une topologie analogue au AHMM où la variable Q_t^D du dernier niveau est liée l'empilement du contexte $Q_{t+1}^{1:D}$ à l'instant suivant.

Dans une même optique et afin d'améliorer la performance du traitement en temps réel, un algorithme d'inférence approximé tel que le filtre à particules pourrait être implanté.

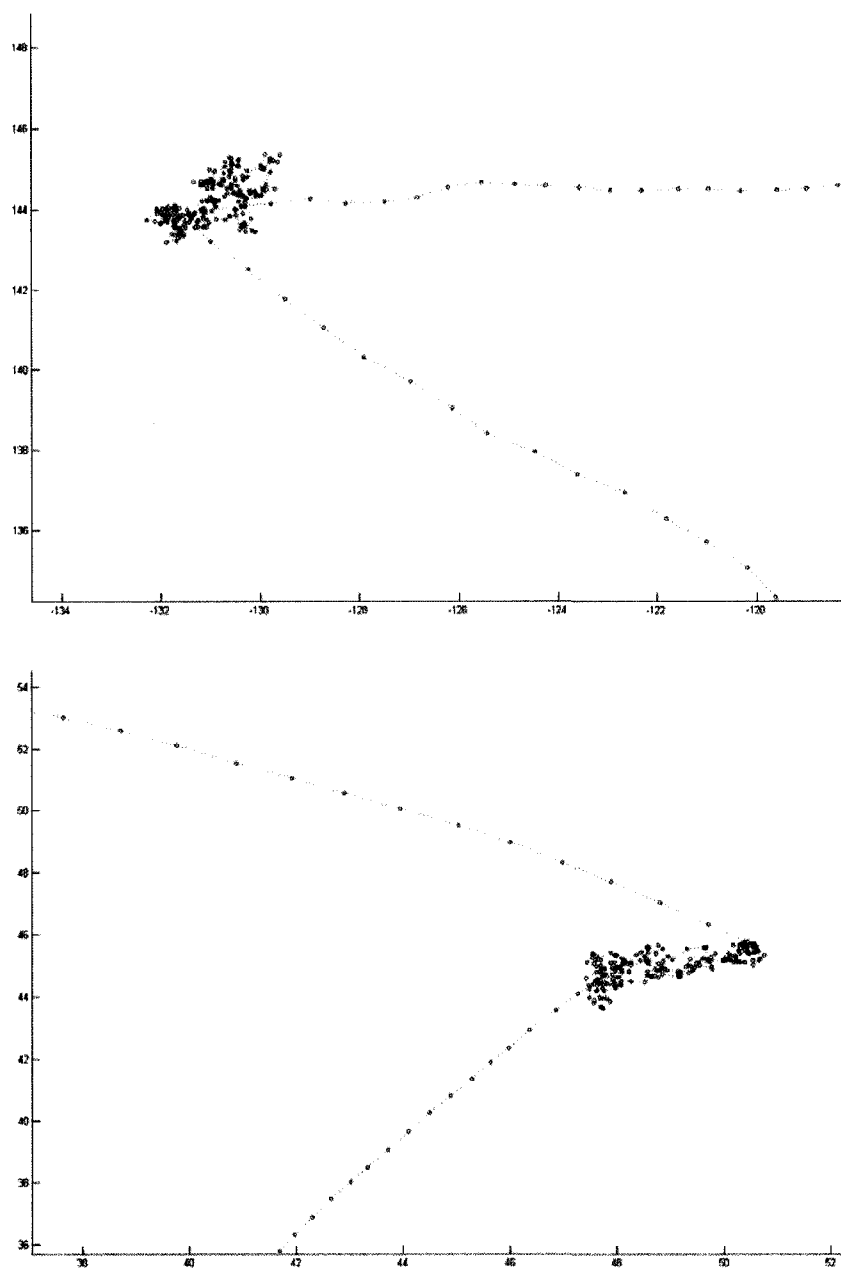


Figure 8.1 : (haut) Représentation d'une trajectoire issue du patron 3-1-4 (« rectiligne accélérée », « immobile », « rectiligne décélérée ») avec un agrandissement sur la primitive centrale. (bas) Représentation d'une trajectoire issue du patron 5-1-6 (« curviligne », « immobile », « curviligne ») avec un agrandissement sur la primitive centrale.

Étant donné que le modèle de segmentation en-ligne fournit une information sur le commencement et la fin des primitives, un modèle hybride HHMM dont la topologie est modifiée pour que les variables $F_{1:T}^D$ du dernier niveau soient observées pourrait bénéficier des avantages du HHMM et du NBC+HHMM.

Le modèle d'observation (gaussien) du PGM et/ou du classificateur en entrée (classificateur+HHMM analogue au NBC+HHMM) pourrait considérer uniquement l'espace de dimensionnalité réduite fourni par PCA/PPCA

Finalement, des développements devraient être considérés afin d'intégrer les diverses considérations énoncées dans l'introduction, notamment, les autres couches du modèle multi-résolution, les interactions entre les agents, l'information sur l'environnement et les périodes récurrentes.

CONCLUSION

En résumé, le but de cette recherche consistait à élaborer un modèle d'activité capable d'analyser en temps réel des trajectoires bidimensionnelles effectuées par un agent mobile autonome sur les bases des modèles graphiques probabilistes. Un HHMM a été étudié et testé dans trois contextes d'utilisation afin d'évaluer sa capacité à reconnaître les patrons et les sous-patrons de trajectoires bidimensionnelles de déplacement. Un modèle de segmentation en-ligne basé sur PPCA a aussi été utilisé pour réduire la charge de calcul sur le HHMM. Les résultats montrent qu'un HHMM entraîné avec l'algorithme de maximisation de l'espérance de façon non supervisée reconnaît une séquence de primitives et détecte l'existence de sous-patrons. Le coût de calcul élevé limite toutefois l'utilisation de tels PGMs en temps réel surtout lorsque la topologie de la hiérarchie se complexifie. L'ajout d'un modèle PPCA de segmentation en-ligne a permis de réduire le coût moyen de calcul du HHMM d'un facteur 10^2 .

Le modèle d'activité obtenu semble être une base intéressante pour d'éventuels développements sur les stratégies de planification, la prédiction de trajectoires et la détection d'anomalies. Des améliorations peuvent notamment être apportées sur la topologie du PGM et l'algorithme d'inférence. Des considérations d'importance telles que les diverses résolutions spatiales et les interactions entre agents mobiles devraient être abordées dans un futur rapproché.

RÉFÉRENCES

- Baker, C., Tenenbaum, J., & Saxe, R. (2006). Bayesian models of human action understanding. *Advances in neural information processing systems*(18), 99-106.
- Barbic, J., Safonova, A., Pan, J. Y., Faloutsos, C., Hodgins, J., & Pollard, N. (2004). *Segmenting motion capture data into distinct behaviors*. Paper presented at the Graphics Interface
- Bengio, Y. (1999). Markovian Models for Sequential Data. *Neural Computing Surveys* 2, 129-162.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*: Oxford University Press.
- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. In *Collection Information science and statistics* (pp. 359-418).
- Bui, H., Venkatesh, S., & West, G. (2001). Tracking and surveillance in wide-area spatial environments using the abstract hidden Markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 177-195.
- Bui, H., Venkatesh, S., & West, G. (2002). Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research*, 17(December), 451-499.
- Bui, H. H. (2003). *A general model for online probabilistic plan recognition*. Paper presented at the Eighteenth International Joint Conference on Artificial Intelligence
- Bui, H. H., Phung, D. Q., & Venkatesh, S. (2004). *Hierarchical hidden Markov models with general state hierarchy*. Paper presented at the Nineteenth National Conference on Artificial Intelligence, San Jose, California.

- Ghahramani, Z. (2001). An Introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 9-42.
- Hu, W., Xie, D., & Tan, T. (2004). A Hierarchical Self-Organizing Approach for Learning the Patterns of Motion Trajectories. *IEEE Transactions on Neural Networks*, 15(1).
- Ihler, A., & Smyth, P. (2006). *Learning Time-Intensity Profiles of Human Activity using Non-Parametric Bayesian Models*. Paper presented at the Neural Information Processing Systems Conference.
- Kaminka, G., Avrahami, D. (2004). *Symbolic behavior-recognition* Paper presented at the Workshop on Modeling Other Agents from Observations.
- Kaminka, G., Fidanboyly, M., Chang, A., & Veloso, M. (2003). Learning the sequential coordinated behavior of teams from observation. In G. Kaminka, Lima, P., Rojas, R. (Ed.), *RoboCup 2002: Robot Soccer World Cup VI* (Vol. 2752, pp. 111-125).
- Kang, C. H., Hwang, J. R., & Li, K. J. (2006). *Trajectory Analysis for Soccer Players*. Paper presented at the 6th IEEE International Conference on Data Mining (ICDM), Hong Kong, China.
- Kwok, C., & Fox, D. (2005). Map-based Multiple Model Tracking of a Moving Object. In S. Berlin (Ed.), *RoboCup 2004: Robot Soccer World Cup VIII* (Vol. 3276, pp. 18-33).
- Lattner, A. D., Miene, A., Visser, U., & Herzog, O. (2006). Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer In S. B. Heidelberg (Ed.), *RoboCup 2005: Robot Soccer World Cup IX* (Vol. 4020, pp. 118-129).
- Lee, K. K., & Xu, Y. (2004). *Boundary modeling in human walking trajectory analysis for surveillance*. Paper presented at the IEEE International Conference on Robotics and Automation.

- Li, C., & Biswas, G. (1999). *Findings Behavior Patterns from Temporal Data using Hidden Markov Model based Unsupervised Classification*. Paper presented at the International ICSC Symposium on Advances in Intelligent Data Analysis (AIDA).
- Liao, L., Fox, D., & Kautz, H. (2004). *Learning and inferring transportation routines*. Paper presented at the National Conference on Artificial Intelligence.
- Luhr, S., Bui, H. H., Venkatesh, S., & West, G. (2003). *Recognition of human activity through hierarchical stochastic learning*. Paper presented at the IEEE International Conference on Pervasive Computing and Communication.
- Murphy, K. P. (2001). The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33.
- Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation, University of California, Berkeley.
- Murphy, K. P., & Paskin, M. (2001). *Linear time inference in hierarchical HMMs*. Paper presented at the NIPS.
- Naïm, P. (2004). *Réseaux bayésiens, 2e édition*. Paris.
- Nguyen, N., Phung, D., Venkatesh, S., & Bui, H. (2005). *Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model*. Paper presented at the IEEE International Conference on Computer Vision and Pattern Recognition
- Nguyen, N. T., Bui, H. H., Venkatesh, S., & West, G. (2003). *Recognizing and monitoring highlevel behaviours in complex spatial environments*. Paper presented at the IEEE International Conference on Computer Vision and Pattern Recognition
- Osentoski, S., Manfredi, V., & Mahadevan, S. (2004). *Learning hierarchical models of activity*. Paper presented at the IEEE/RSJ International Conference on Robots and Systems

- Plamondon, R., & Srihari, S. N. (2000). On-line and Off-line Handwriting Recognition a Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1).
- Pynadath, D. V. (1999). *Probabilistic grammars for plan recognition.*, University of Michigan.
- Pynadath, D. V., & Wellman, M. P. (1995). *Accounting for context in plan recognition, with application to traffic monitoring.* Paper presented at the Eleventh Conference on Uncertainty in Artificial Intelligence, San Francisco.
- Pynadath, D. V., & Wellman, M. P. (2000). *Probabilistic statedependent grammars for plan recognition.* Paper presented at the Sixteenth Conference of Uncertainty in Artificial Intelligence San Francisco.
- Riley, P. (2005). *Coaching: Learning and Using Environment and Agent Models for Advice.* Carnegie Mellon University, Pittsburgh.
- Riley, P., & Veloso, M. (2002). Recognizing probabilistic opponent movement models. In S. Verlag (Ed.), *RoboCup-2001: Robot Soccer World Cup V* (Vol. 2377, pp. 453-458).
- Riley, P., & Veloso, M. (2004). *Advice Generation from Observed Execution: Abstract Markov Decision Process Learning.* Paper presented at the Nineteenth National Conference on Artificial Intelligence.
- Robertson, N., & Reid, I. (2005). *Behavior Understanding in Video: A Combined Method.* Paper presented at the Tenth IEEE International Conference on Computer Vision
- Roweis, S. (1998). *EM Algorithms for PCA and SPCA.* Paper presented at the Advances in Neural Information Processing Systems, Denver, Colorado.
- Roweis, S. T., & Salakhudinov, R. R. (2003). Simultaneous Localization and Surveying with Multiple Agents. In *Essential C# Fast.*

- Saria, S., & Mahadevan, S. (2004). *Probabilistic plan recognition in multiagent systems*. Paper presented at the International Conference on Artificial Intelligence and Planning Systems.
- Smith, T. (2007). *Probabilistic Planning for Robotic Exploration*. Carnegie Mellon Pittsburgh, Pennsylvania
- Sukthankar, G., & Sycara, K. (2005). *A Cost Minimization Approach to Human Behavior Recognition*. Paper presented at the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems.
- Sukthankar, G., & Sycara, K. (2006). *Robust Recognition of Physical Team Behaviors using Spatio-Temporal Models*. Paper presented at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Hakodate, Japan.
- Tipping, M. E., & Bishop, C. M. (1999). Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, 61, Part 3, 611-622.
- Vlachos, M., Kollios, G., & Gunopulos, D. (2002). *Discovering similar multidimensional trajectories*. Paper presented at the 18th International Conference on Data Engineering.
- Yin, J., Shen, D., Yang, Q., & Li, Z. E. (2005). *Activity Recognition through Goal-Based Segmentation*. Paper presented at the Twentieth National Conference on Artificial Intelligence, Pittsburgh.
- Yin, M., Chai, X., & Yang, Q. (2004). *High-level goal recognition in a wireless lan*. Paper presented at the National Conference on Artificial Intelligence San Jose, California.

ANNEXE 1

Concepts de base sur les réseaux bayésiens

Les concepts de base derrière les BNs, tirés de Naïm (2004) et de Bishop (2006), sont sommairement présentés ici afin de mettre en évidence le lien direct qui existe entre la topologie d'un BN et le calcul d'inférence pouvant être fait sur le réseau.

Tout d'abord, on dit qu'il y a « d-séparation » entre deux nœuds d'un BN s'il n'existe pas de chemin causal entre les deux nœuds et qu'il n'existe aucune évidence rendant les deux nœuds corrélés (*explaining away*).

Un BN forme un « réseau complètement connecté » si toutes les paires de nœuds sont connectés entre elles et ce, sans égard à l'orientation des arêtes.

À l'opposé, un BN forme un « réseau singulièrement connecté » ou un « poly-arbre » (*polytree*) s'il existe uniquement un chemin entre chaque paire de nœuds du BN et ce, sans égard à l'orientation des arêtes. Ainsi, si un BN contient au moins un « cycle (non orienté) », alors le BN ne forme pas un « réseau singulièrement connecté ».

Le calcul d'inférence (exact) est fait avec l'algorithme de propagation de la croyance par messages locaux qui s'avère approprié lorsque la structure du BN est un poly-arbre puisque la d-séparation entre deux ensembles de nœuds est alors facile à évaluer. Dans le cas contraire, un « arbre de jonction » doit être construit. Le BN doit être converti en un « graphe non orienté » ou « réseau de Markov » ou « graphe moral ». C'est la « moralisation ». Cette étape implique la formation de « cliques », c'est-à-dire des plus gros sous-graphes complètement connectés. Finalement, le graphe non orienté est « triangulé » puis certaines arêtes sont retirées pour obtenir l'arbre de jonction.

L'algorithme de propagation de la croyance permet alors d'évaluer la relation (1.1) à partir des distributions de probabilité conditionnelles (CPD) définies pour chaque variable. Dans le cas d'une variable continue, il est

souvent question de la distribution gaussienne, de un, parce qu'il s'agit d'une distribution appartenant à la famille exponentielle et, de deux, parce qu'en conditionnant sur d'autres variables (discrète et/ou continue), la CPD devient gaussienne conditionnelle, gaussienne linéaire ou gaussienne linéaire conditionnelle permettant du coup de modéliser n'importe quelles formes de distribution moyennant un nombre suffisant de degrés de liberté. Dans le cas d'une variable discrète, les paramètres se résument aux probabilités de la table de probabilité conditionnelle (CPT).

ANNEXE 2

Analyse en composantes principales (PCA)

Forme particulière de réduction de la dimensionnalité où l'on recherche une variété affine qui passe le plus près des données, i.e. une transformation linéaire réduisant la dimension M à M' ($M' < M$) tout en minimisant la perte d'information.

Soit $\mathbf{x}^n, n=1, \dots, N$ une donnée $\mathbf{x} \in \mathbb{R}^M$ avec des coordonnées (x_1, x_2, \dots, x_M) . On cherche à exprimer \mathbf{x} à l'aide de coordonnées plus simples $(a_1, a_2, \dots, a_{M'})$, $M' < M$. Cela peut être fait en toute généralité de la façon suivante

$$\mathbf{x} = \sum_{i=1}^M a_i \mathbf{u}_i$$

où les $a_i, i=1, \dots, M$ sont les coordonnées selon chacun des vecteurs \mathbf{u}_i formant une base vectorielle orthonormale, i.e. $\mathbf{u}_i \cdot \mathbf{u}_j = \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$. En conservant uniquement un sous-ensemble $M' < M$ des vecteurs de base et en remplaçant les coordonnées supplémentaires par une constante, il devient possible d'approximer \mathbf{x} par

$$\hat{\mathbf{x}} = \sum_{i=1}^{M'} a_i \mathbf{u}_i + \sum_{i=M'+1}^M b_i \mathbf{u}_i \quad (9.1)$$

Dans la base appropriée (le détail de la base minimisant la perte d'information est exposé plus bas), cette approximation représente les données sur la variété affine et les $a_i, i=1, \dots, M'$ sont appelées les coordonnées réduites. On calcule leur valeur en minimisant la somme des erreurs au carré

$$\begin{aligned}
E &= \frac{1}{2} \sum_{n=1..N} \|\mathbf{x}^n - \hat{\mathbf{x}}^n\|^2 \\
&= \frac{1}{2} \sum_{n=1}^N (\mathbf{x}^n - \hat{\mathbf{x}}^n)^T (\mathbf{x}^n - \hat{\mathbf{x}}^n) \\
&= \frac{1}{2} \sum_{n=1}^N \left(\sum_{i=1}^M a_i^n - \sum_{i=M'+1}^M (a_i^n + b_i) \right) \left(\sum_{j=1}^M a_j^n - \sum_{j=M'+1}^M (a_j^n + b_j) \right) \mathbf{u}_i^T \mathbf{u}_j \\
&= \frac{1}{2} \sum_{n=1}^N \sum_{i,j=M'+1}^M (a_i^n - b_i)(a_j^n - b_j) \mathbf{u}_i^T \mathbf{u}_j \\
&= \frac{1}{2} \sum_{n=1}^N \sum_{i=M'+1}^M (a_i^n - b_i)^2
\end{aligned}$$

de sorte que

$$\begin{aligned}
\frac{\partial E}{\partial b_i} &= \frac{1}{2} \frac{\partial}{\partial b_i} \sum_{n=1}^N \sum_{i=M'+1}^M (a_i^n - b_i)^2 \\
&= - \sum_{n=1}^N \sum_{i=M'+1}^M (a_i^n - b_i) \\
&= - \sum_{n=1}^N \sum_{i=M'+1}^M a_i^n + N \sum_{i=M'+1}^M b_i \\
&= 0
\end{aligned}$$

En isolant b_i , on obtient le résultat suivant

$$\begin{aligned}
N \sum_{i=M'+1}^M b_i &= \sum_{n=1}^N \sum_{i=M'+1}^M a_i^n \\
b_i &= \frac{1}{N} \sum_{n=1}^N a_i^n
\end{aligned} \tag{9.2}$$

où b_i est interprétable comme la i ème coordonnée de la moyenne des données. En effet,

$$\begin{aligned}
\sum_{i=1}^M b_i \mathbf{u}_i &= \sum_{i=1}^M \frac{1}{N} \sum_{n=1}^N a_i \mathbf{u}_i \\
&= \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^M a_i \mathbf{u}_i \\
&= \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n \\
&= \bar{\mathbf{x}}
\end{aligned}$$

d'où

$$b_i = \bar{\mathbf{x}}^T \mathbf{u}_i \quad (9.3)$$

Ainsi, le second terme de l'expression (9.1) correspond aux $M - M'$ composantes de la moyenne des données. En substituant (9.3) dans l'expression (9.2), on obtient le résultat suivant

$$\begin{aligned}
\frac{1}{N} \sum_{n=1}^N a_i^n &= \bar{\mathbf{x}}^T \mathbf{u}_i \\
\sum_{n=1}^N a_i^n &= N \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n{}^T \mathbf{u}_i
\end{aligned}$$

d'où

$$a_i = \mathbf{x}^T \mathbf{u}_i \quad (9.4)$$

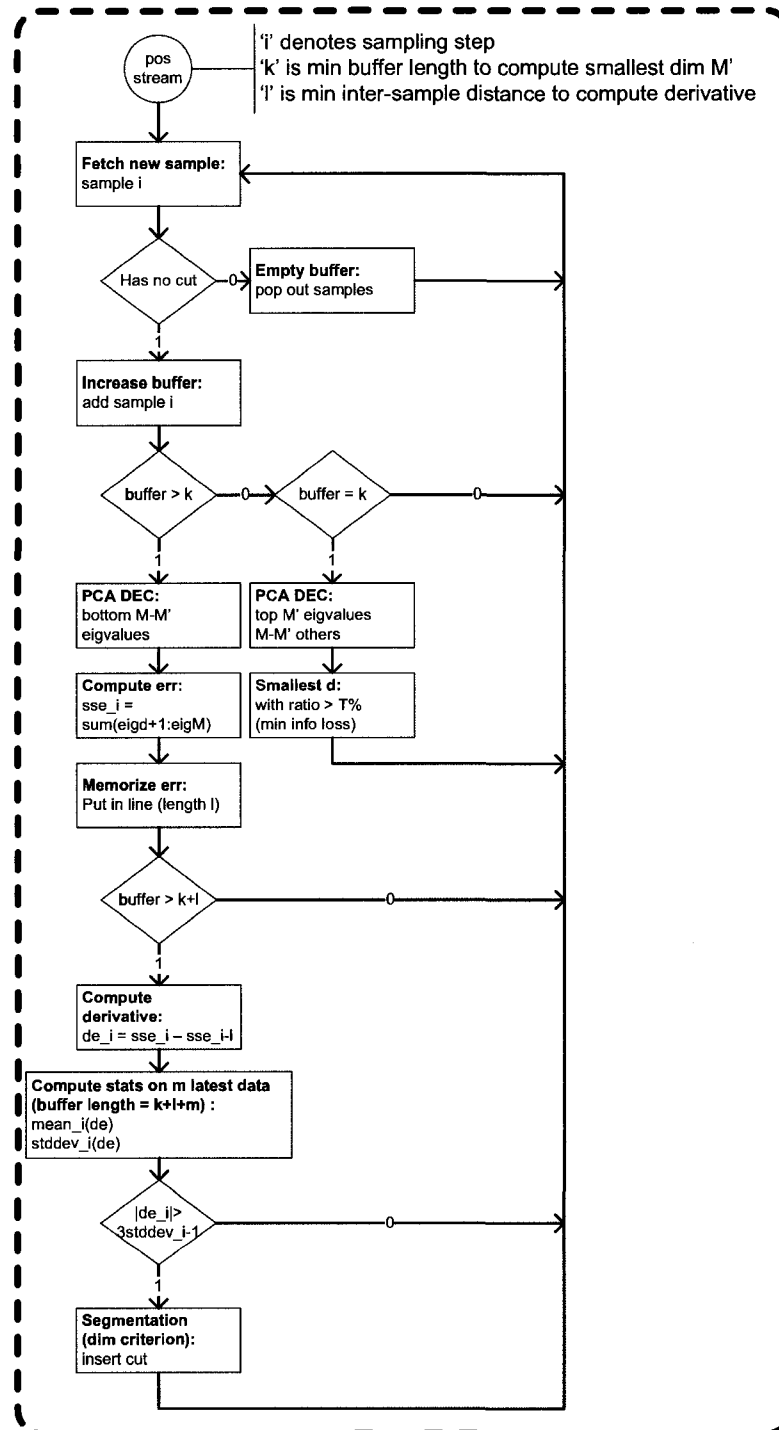
Ce résultat rappelle l'affirmation initiale comme quoi les a_i représentent les coordonnées de \mathbf{x} dans la base orthonormale des \mathbf{u}_i . Cette projection des données dans une autre base s'interprète comme une rotation du système de coordonnées. En substituant les expressions (9.3) et (9.4) dans la définition de la somme des erreurs au carré, on obtient

$$\begin{aligned}
E &= \frac{1}{2} \sum_{n=1}^N \sum_{i=M'+1}^M (\mathbf{x}^n \mathbf{u}_i - \bar{\mathbf{x}} \mathbf{u}_i)^2 \\
&= \frac{1}{2} \sum_{n=1}^N \sum_{i=M'+1}^M ((\mathbf{x}^n - \bar{\mathbf{x}})^T \mathbf{u}_i)^2 \\
&= \frac{1}{2} \sum_{i=M'+1}^M \sum_{n=1}^N \mathbf{u}_i^T (\mathbf{x}^n - \bar{\mathbf{x}}) (\mathbf{x}^n - \bar{\mathbf{x}})^T \mathbf{u}_i \\
&= \frac{1}{2} \sum_{i=M'+1}^M \mathbf{u}_i^T \left(\sum_{n=1}^N (\mathbf{x}^n - \bar{\mathbf{x}}) (\mathbf{x}^n - \bar{\mathbf{x}})^T \right) \mathbf{u}_i \\
&= \frac{1}{2} \sum_{i=M'+1}^M \mathbf{u}_i^T \Sigma \mathbf{u}_i
\end{aligned} \tag{9.5}$$

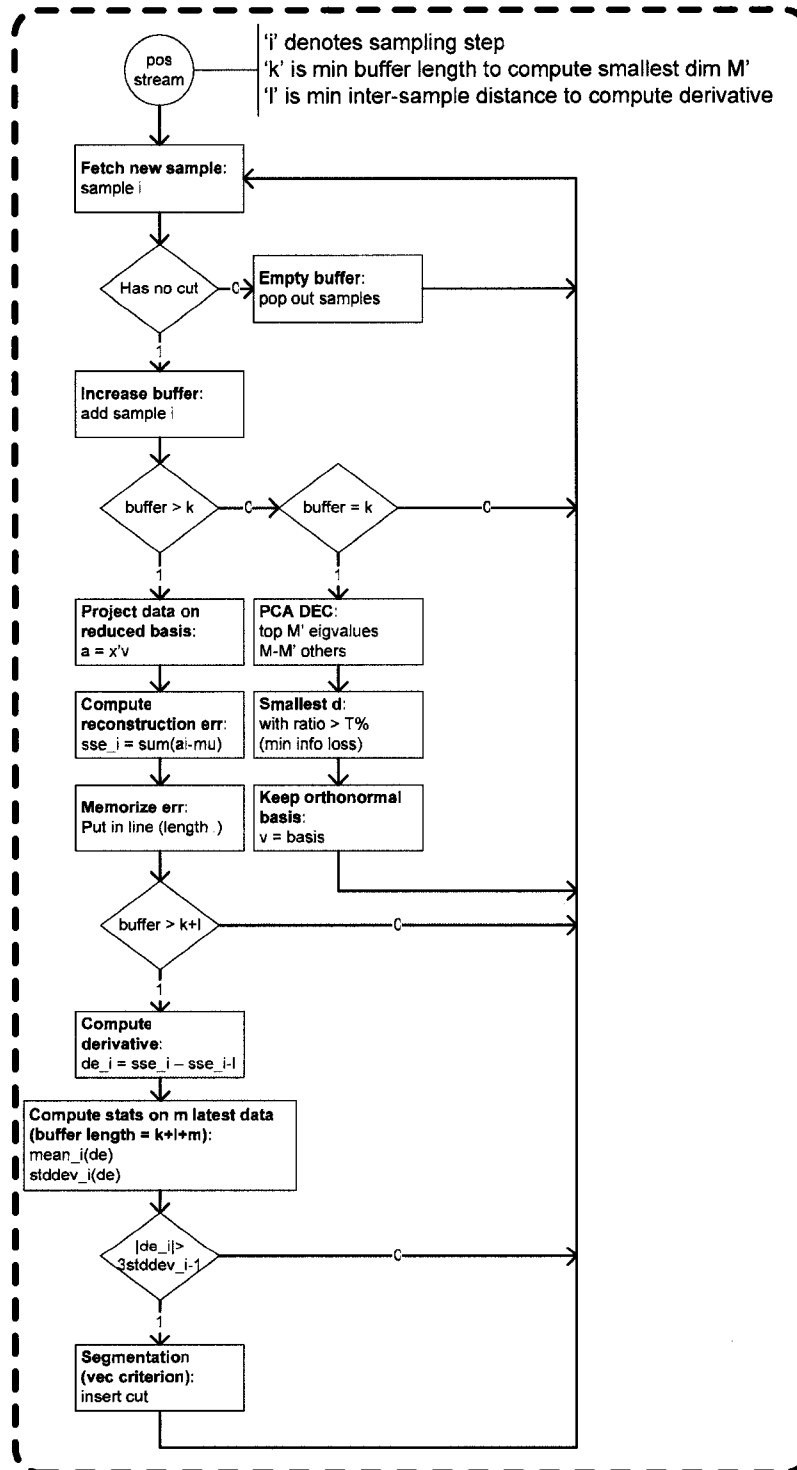
où Σ est la matrice de covariance des données \mathbf{x}^n . Il est démontré que les \mathbf{u}_i qui minimisent (9.5) correspondent aux vecteurs propres de ladite matrice de covariance (Bishop, 1995). Il en résulte que la somme des erreurs au carré s'exprime en terme des $M - M'$ plus petites valeurs propres λ_i de cette même matrice.

$$E = \frac{1}{2} \sum_{i=M'+1}^M \lambda_i \tag{9.6}$$

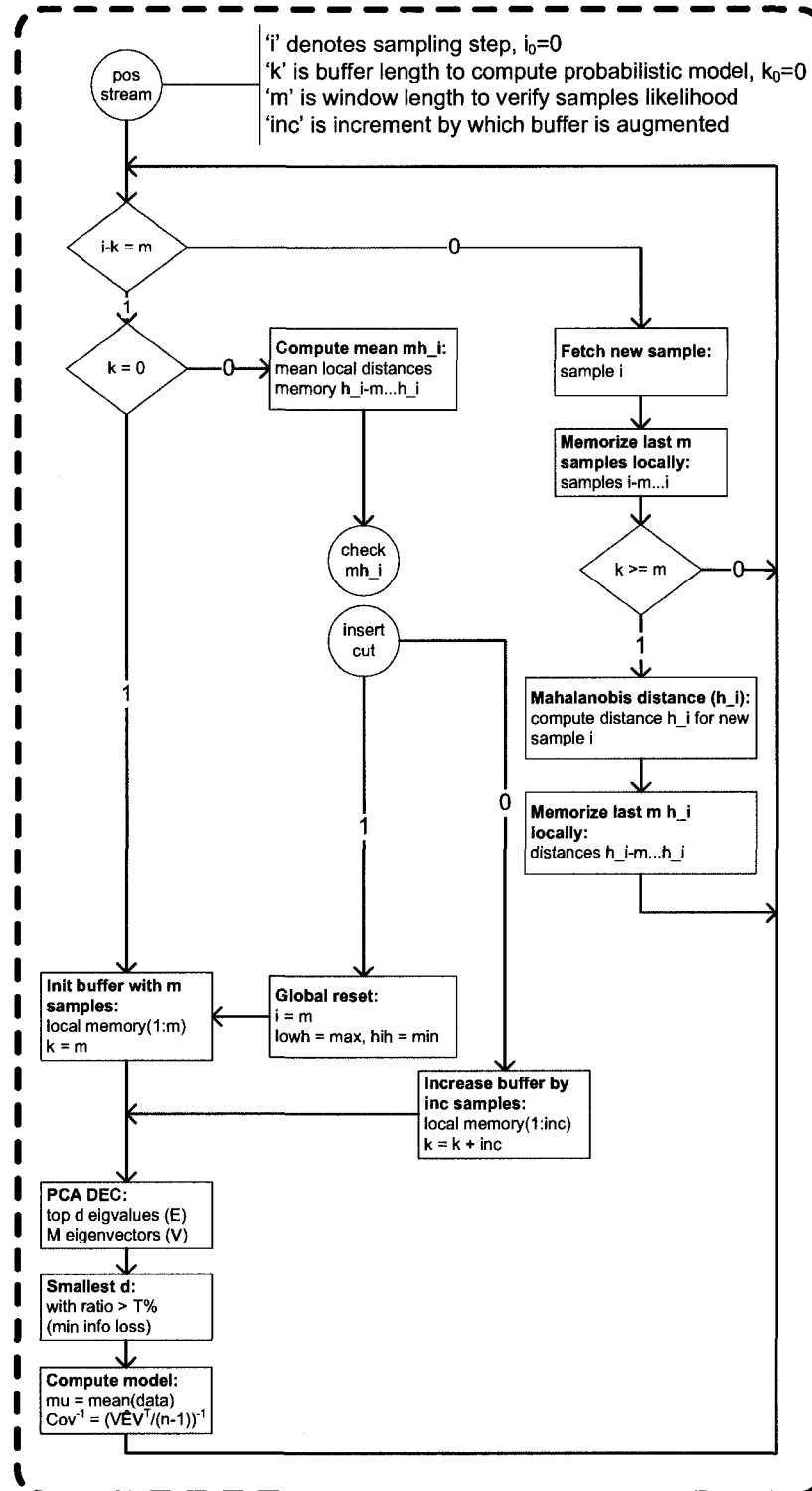
Technique de segmentation PCA (critère dimensionnalité)

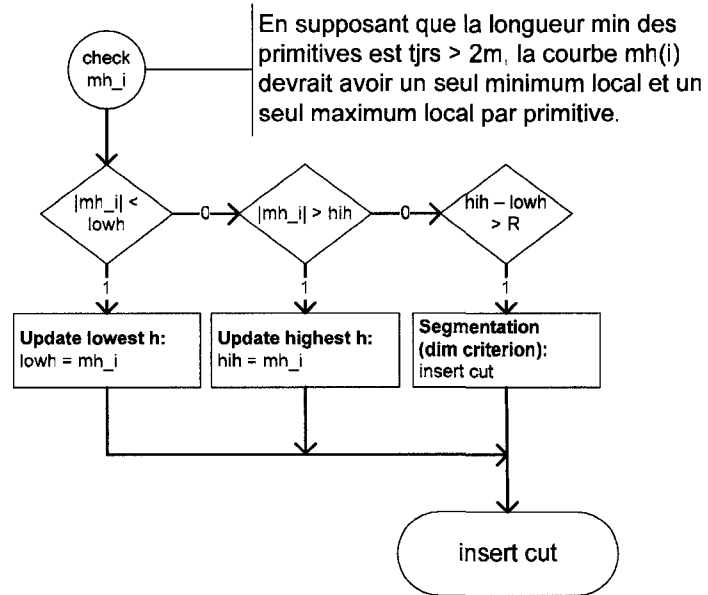


Technique de segmentation PCA (critère base vectorielle)



Technique de segmentation PPCA





Définition des modèles de segmentation en-ligne en terme de leurs hyper-paramètres

Les hyper-paramètres sont énoncés afin de correspondre exactement à la formulation dans le module Simulink de Matlab.

PCA dim-x : xième modèle basé sur la technique de segmentation PCA avec le critère de dimensionnalité;

PCA vec-x : xième modèle basé sur la technique de segmentation PCA avec le critère de la base vectorielle;

PPCA-x : xième modèle basé sur la technique de segmentation PPCA;

(tableau à la page suivante)

Modèles :	k (>Nsig)	L	m	maxlength	tol (%)	kstddev
PCA dim – 30	35	8	1	500	95	3
PCA dim – 33	80	20	1	500	95	2,6
PCA dim – 34	35	8	1	500	95	2,6
PCA dim – 35	35	8	1	500	95	2,6
PCA dim – 101	35	8	1	500	99	3
PCA dim – 108	35	8	1	500	99	2,6
PCA dim – 114	35	5	1	500	99	3
PCA dim – 124	35	5	1	500	99	2,6
modèle	k (>Nsig)	L	m	maxlength	tol (%)	kstddev
PCA vec – 38	80	20	1	500	95	2,6
PCA vec – 52	80	10	1	500	95	2,6
PCA vec – 53	80	10	1	500	95	2,6
PCA vec – 54	80	10	1	500	95	2,6
PCA vec – 55	80	10	1	500	90	2,6
PCA vec – 83	60	8	1	500	95	2,6
PCA vec – 88	60	8	1	500	99	2,6
PCA vec – 97	35	8	1	500	95	2,6
PCA vec – 104	35	8	1	500	99	2,6
PCA vec – 114	35	5	1	500	95	2,6
PCA vec – 115	35	5	1	500	90	2,6
modèle	m (>Nsig)	inc (~0,1m)	-	maxlength	tol (%)	R
PPCA – 11	35	5	NA	500	95	100
PPCA – 12	35	5	NA	500	95	100
PPCA – 24	60	6	NA	500	95	100
PPCA – 28	35	5	NA	500	95	100

ANNEXE 3**Schéma Simulink des opérations de prétraitement sur la
dimensionnalité**

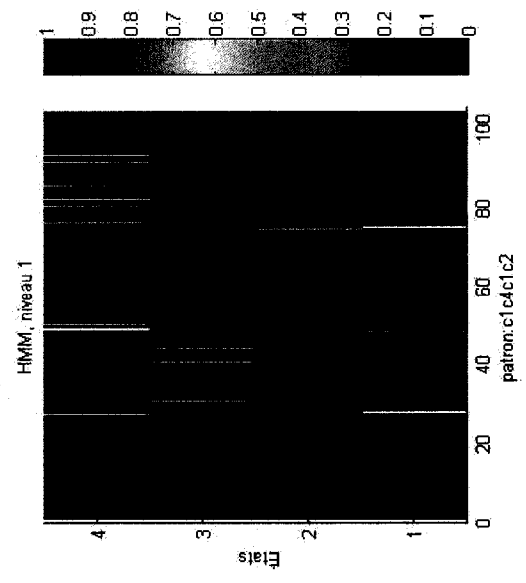
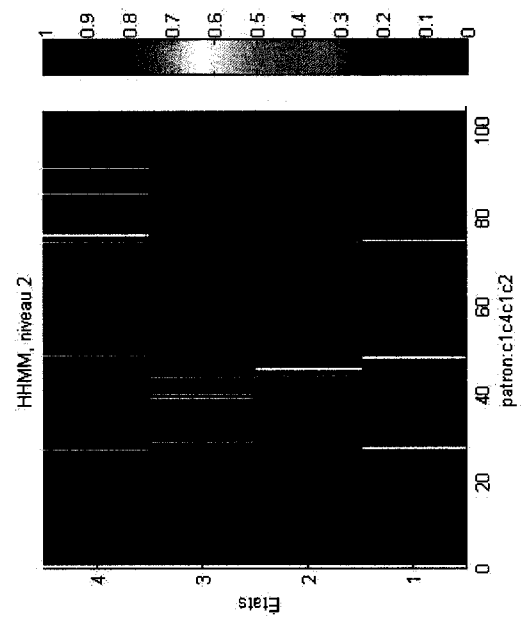
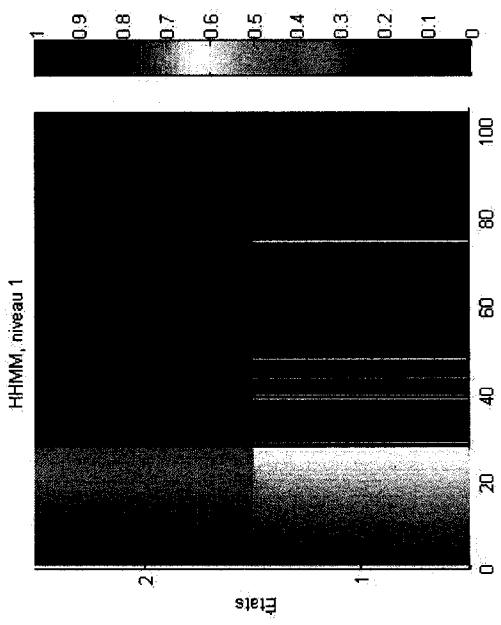
ANNEXE 4

Validation du PGM seul : résultats complémentaires

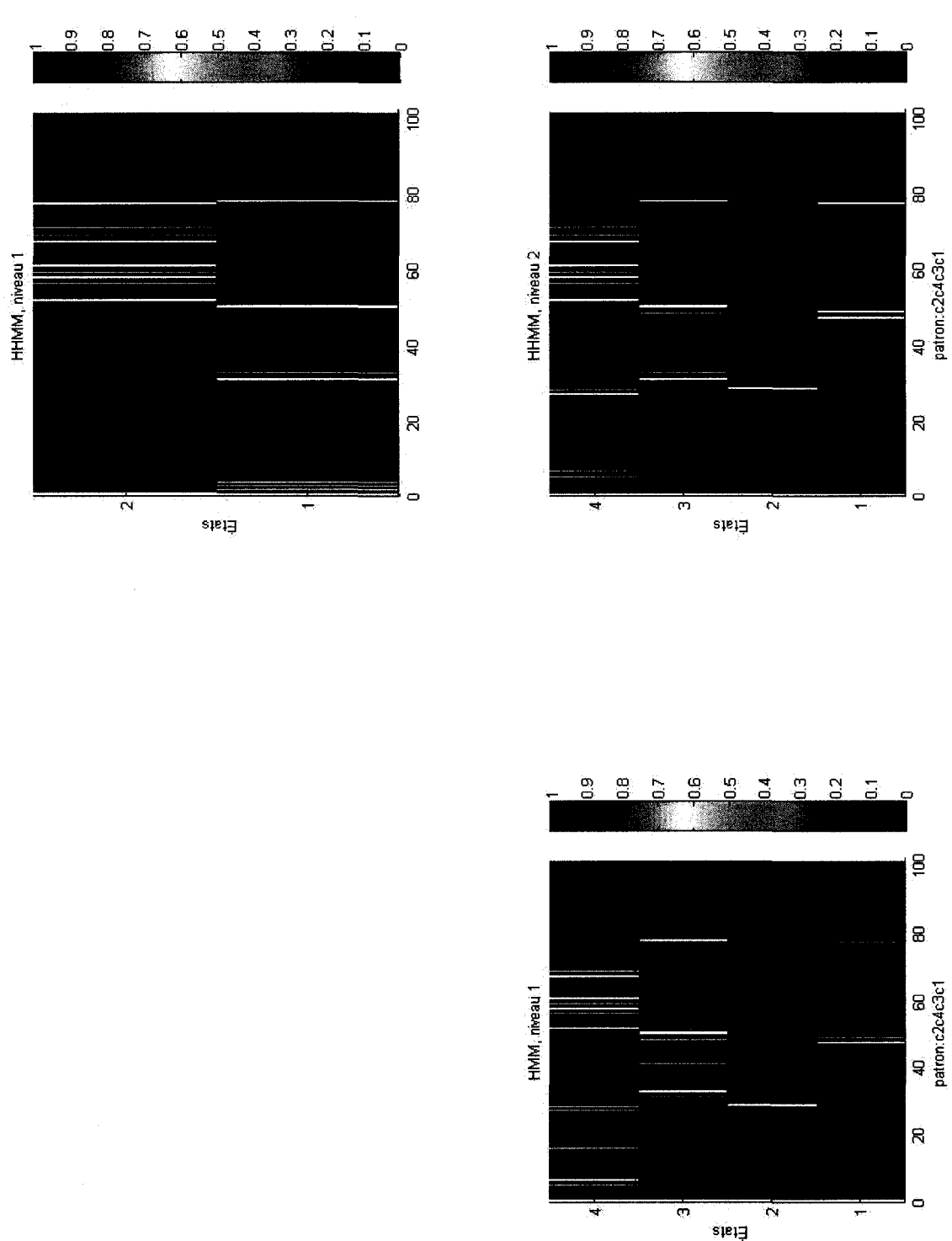
Cette annexe contient quatre des cinq représentations de la probabilité marginale de chaque état dans l'empilement du contexte de chaque PGM testé au chapitre 6 pour des observations dans un espace de primitives gaussiennes à 3 dimensions.

Les colonnes de la figure représentent les divers niveaux d'un même modèle. Chaque graphe de la figure représente la topologie du modèle marginalisée (i.e. « figée ») étant donné une certaine séquence d'observations. L'indice temporel est en abscisse, les états possibles, en ordonnée et les probabilités sont indiquées par les couleurs.

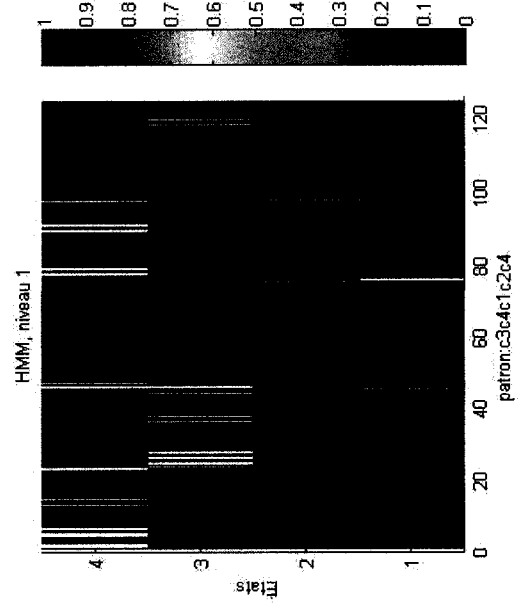
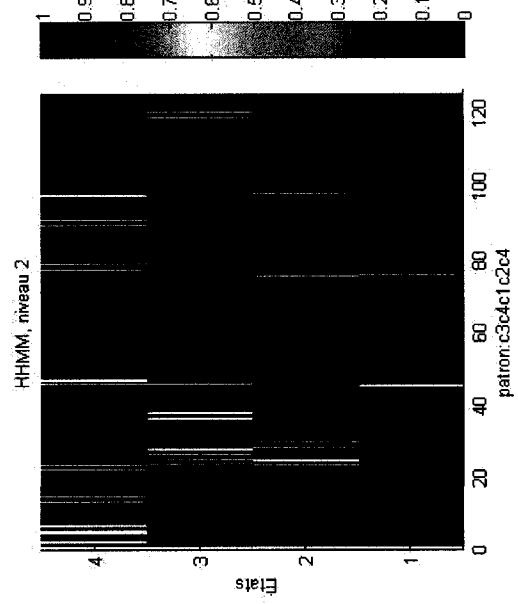
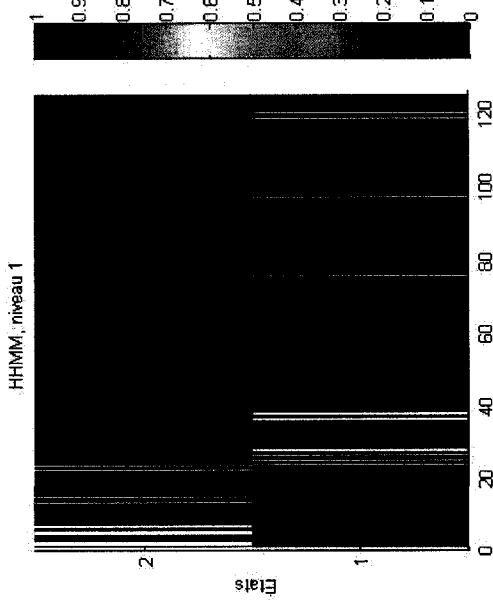
Séquence d'observations (1-4-1-2).



Séquence d'observations (2-4-3-1).



Séquence d'observations (3-4-1-2-4).



Séquence d'observations (4-1-2).

